



**Universidad Nacional Mayor de San Marcos**  
**Universidad del Perú. Decana de América**  
Facultad de Ingeniería de Sistemas e Informática  
Escuela Académico Profesional de Ingeniería de Sistemas

**Desarrollo de una metodología para la elicitación y  
análisis del proceso de la ingeniería de requerimientos**

**TESINA**

Para optar el Título Profesional de Ingeniero de Sistemas

**AUTORES**

Rosa María JURUPE GARCÍA

Paola Aurelia PACHECO RAMÍREZ

**ASESOR**

José César PIEDRA ISUSQUI

Lima, Perú

2007



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

## Referencia bibliográfica

---

Jurupe, R. (2007). *Desarrollo de una metodología para la elicitación y análisis del proceso de la ingeniería de requerimientos*. Tesina para optar el título de Ingeniero de Sistemas. Escuela Académico Profesional de Ingeniería de Sistemas, Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima, Perú.

---

### **DEDICATORIA**

A Dios, por su eterno amor, su sabiduría y permitirme  
ver mis sueños hacerse realidad.  
Rosa María Jurupe García

### **DEDICATORIA**

A mi madre, por su constante apoyo y amor.  
Paola Aurelia Pacheco Ramírez

## **RESUMEN**

### **DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

Rosa María Jurupe García  
Paola Aurelia Pacheco Ramírez

**Septiembre – 2007**

<b>Asesor</b>	<b>:</b>	<b>Ing. José Piedra Isusqui</b>
<b>Grado</b>	<b>:</b>	<b>Bachiller</b>

---

La base de un proyecto de desarrollo software lo constituyen los requerimientos que soportan su estructura y definen su dinámica. Para garantizar que el producto final satisfaga las necesidades y expectativas de los usuarios se debe construir con requerimientos tomados de los ambientes real del negocio, los cuales sean consistentes y actuales.

La obtención de requerimientos es la fase de la cual depende en gran medida el grado de éxito del proyecto, por tal razón es necesario aplicar un método flexible, iterativo y orientado en el usuario, el cual haga de un conjunto de técnicas y herramientas que faciliten la elicitación, análisis, especificación y validación de los requerimientos del producto.

En este documento se presenta a la obtención de requerimientos como una solución a algunas causas de los fracasos de los proyectos de software, reduciendo gran parte de su sobre costo, permitiendo la estimación de tiempo y esfuerzo acordes a la realidad, y facilitando la elaboración de un producto con calidad.

## **ABSTRACT**

### **DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

Rosa María Jurupe García

Paola Aurelia Pacheco Ramírez

**September – 2007**

**Adviser** : **Ing. José Piedra Isusqui**

**Degree** : **Bachelor**

---

The basis of a Project of development / software is constituted by the requests that support its structure and define its dynamics. To guarantee that the final product satisfies the needs and the users' expectations, it must build with requirements taken from the real business environment witch they must be consistent and currents.

The project's success depends on the Requests Obtaining phase for this reason, it's necessary to apply a flexible, iterative method oriented in the user, who wills user a set of techniques and tools that facilitates the extraction, analysis, specification and validation of the product's request.

In this document the request obtaining is Shawn as solution to some failures causes from the software's projects reducing overcost, allowing the estimation of time and effort, according to the reality and facilitating the elaboration of a product with quality.

## INDICE

RESUMEN .....	4
INTRODUCCIÓN.....	11
1. PLANTEAMIENTO DEL PROBLEMA.....	13
1.1 Descripción de la Realidad .....	13
1.2 Antecedentes y Formulación del Problema .....	17
1.3 Importancia y Justificación.....	21
1.4 Definición del Problema.....	23
1.5 Limitaciones y Alcances .....	24
1.6 Variantes del Problema .....	25
2. OBJETIVOS.....	26
2.1 Objetivo General .....	26
2.2 Objetivo Específicos.....	26
3. MARCO TEORICO CONCEPTUAL .....	27
3.1 Antecedentes de la investigación.....	27
3.2 Bases Teóricas .....	28
3.2.1 La Ingeniería de Software.....	28
3.2.2 El proceso de ingeniería de software.....	28
3.2.3 El proceso de desarrollo de software.....	28
3.2.4 La necesidad de la Ingeniería de Requerimientos.....	29
3.2.5 Ingeniería de Requerimientos .....	30
3.2.6 Importancia de la ingeniería de requerimientos.....	32
3.2.7 Requerimientos .....	33
3.2.7.1 Tipos de Requerimientos.....	33
3.2.7.2 Dificultades para definir los requerimientos .....	36
3.2.7.3 Propiedades deseables de los requerimientos .....	37
3.2.8 ¿Para qué un Proceso de Ingeniería de Requerimientos?.....	43
3.2.9 Modelo de proceso de IR.....	43
3.2.10 Problemas con el proceso de Ingeniería de Requerimientos.....	47
3.2.11 Actividades en el Proceso de desarrollo de los requerimientos .....	49
3.2.12 Gestión de Requerimientos .....	54
3.3 Definición de Términos Básicos .....	59
4. METODOLOGIA DE LA INVESTIGACION .....	62
4.1 Propuesta de modelo de procesos de ingeniería de requisitos.....	65
4.1.1 Ciclos de iteración .....	66
4.2 Propuesta de la metodología para la elicitación de requerimientos .....	68
4.2.1 Tarea 1: Formar el equipo multidisciplinario.....	69
4.2.2 Tarea 2: Obtener información sobre el domino del problema y el Sistema Actual	71
4.2.3 Tarea 3: Preparar y realizar las reuniones de elicitación/negociación. ....	73
4.2.4 Tarea 4: Identificar/revisar los objetivos del sistema.....	74
4.2.5 Tarea 5: Clasificar / Revisar Requerimientos .....	75
4.2.6 Tarea 6: Evaluar y racionalizar .....	76



4.2.7	Tarea 7: Dar Prioridad .....	77
4.2.8	Tarea 8: Integrar y Validar.....	78
4.3	Propuesta de la Metodología para el Análisis de Requerimientos .....	80
4.3.1	Tarea 1: Reducir ambigüedades en los requerimientos.....	81
4.3.2	Tarea 2: Traducir a lenguaje técnico los requerimientos. ....	82
4.3.3	Tarea 3: Analizar los requerimientos de almacenamiento de información.....	83
4.3.4	Tarea 4: Analizar los requerimientos funcionales del Sistema.....	84
4.3.5	Tarea 5: Analizar los requerimientos no funcionales del Sistema....	85
4.3.6	Tarea 6: Desarrollar prototipos .....	86
4.4	Documentación de los Requerimientos .....	89
4.4.1	Tarea 1: Documentar Etapa .....	89
4.5	Documento de Requerimientos.....	90
4.5.1	Usuarios del Documento de Requerimientos .....	90
4.5.2	Estructura del Documento de Requerimientos .....	90
4.5.3	Definición de Partes del Documento del Requerimientos .....	92
4.6	Técnicas .....	96
4.6.1	Entrevistas y cuestionarios.....	96
4.6.2	Sistemas existentes.....	98
4.6.3	Grabaciones de video y de audio .....	98
4.6.4	Brainstorming (tormenta de ideas).....	99
4.6.5	Arqueología de documentos.....	100
4.6.6	Aprendiz.....	100
4.6.7	JAD (Joint Application Development/Desarrollo conjunto de aplicaciones): 101	
4.6.8	Etnografía.....	102
4.6.9	Run Use Case WorkShop (Talleres de Trabajo basados en los Casos de Uso) 102	
4.6.10	Prototipos .....	103
4.6.11	Análisis FODA.....	105
4.6.12	Cadena de valor.....	105
4.6.13	Modelo de clase conceptual   Diagrama Conceptual   Diagrama de Clases Conceptual .....	106
4.6.14	Diagrama de pescado (Ishikawa Diagram, Cause-and-Effect o Fishbone Diagram)107	
4.6.15	Glosario .....	108
4.6.16	Diagrama de Actividad .....	109
4.6.17	Casos de uso .....	109
4.6.18	Casa de calidad QFD: Requerimientos versus Casos de Uso.....	110
4.6.19	Checklist (lista de verificación).....	111
4.6.20	Análisis de Factores críticos de éxito .....	111
4.6.21	Análisis comparativo de las técnicas de ingeniería de requerimientos	112
4.7	Plantillas utilizadas en la elicitación y análisis de requerimientos .....	117
4.7.1	Plantilla para los objetivos del sistema .....	118
4.7.2	Plantilla para Requerimientos de almacenamiento de la información....	120
4.7.3	Plantilla para Requerimiento Funcionales .....	121
4.7.4	Plantilla para requerimiento No funcional.....	124

4.7.5	Plantilla para conflictos .....	125
4.7.6	Plantilla para clases .....	127
4.7.7	Plantilla para asociación .....	127
4.8	Caso Práctico: Sistema logístico de una Distribuidora de alimentos .....	129
	Registrar ingreso por compras .....	139
	Traslado entre almacenes .....	141
	Registrar Parte de movimiento de materiales .....	144
	Valorizar Costo Promedio .....	147
	Consulta de stock del material .....	150
4.9	Herramientas automatizadas para la administración de Requerimientos ..	162
4.9.1	RequisitePro .....	162
4.9.2	Doors .....	164
5.	CONCLUSIONES .....	166
6.	RECOMENDACIONES .....	168
7.	REFERENCIAS .....	169
8.	ANEXOS .....	171
8.1	Análisis comparativo de metodologías utilizadas para la obtención de requerimientos y la metodología propuesta .....	171
8.2	Factores críticos de éxito del Proyecto .....	173
8.3	UML .....	173
4.10	Lenguaje restrictivo (constraint) de objetos (OCL) .....	175

## INDICE DE FIGURAS

Figura 1.1 Costo de Corrección Tardía (Boehm, 1981).....	13
Figura 1.2 : Problemas de Desarrollo: Un ejemplo regional(Chile).....	14
Figura 1.3: Resultados del Informe de GAO.....	20
Figura 1.4: Resultados del informe CHAOS.....	21
Figura 3.1 La ingeniería de requerimientos en el ciclo de vida del desarrollo de Software .....	31
Figura 3.2: La Ingeniería de requerimientos como un proceso de Comunicación	32
Figura 3.3: Modelo del Proceso de la Ingeniería de Requerimientos .....	44
Figura 3.4: Modelo Tradicional en Cascada .....	45
Figura 3.5: Modelo del Ciclo de vida en Espiral .....	46
Figura 3.6: Diagrama de actividades del Proceso de la Ingeniería de Requerimientos .....	48
Figura 3.7: Actividad de Validación de Requerimientos .....	54
Figura 3.8: Evolución de los requerimientos.....	55
Figura 3.9: Etapas principales del proceso de administración de cambio .....	58
Figura 4.1: Modelo Propuesto de Proceso de la Ingeniería de Requerimientos..	65
Figura 4.2: Diagrama de Actividades de la actividad de Elicitación del Proceso de la Ingeniería de Requerimientos.....	79
Figura 4.3: Diagrama de Actividades de la actividad de Análisis del Proceso de la Ingeniería de Requerimientos .....	88
Figura 4.4: Las plantillas como elemento de elicitación y negociación.....	118
Figura 4.5: Diagrama de Clases del Modulo de Almacén .....	154
Figura 8.1: Jerarquía de los diagramas UML 2.0 .....	174

## INDICE DE TABLAS

Tabla 1.1 : Relevamiento de proyectos de software en EEUU por el Standish Group. ....	14
Tabla 1.2 Causa percibida de éxito y fracaso de un proyecto. ....	15
Tabla 1.3 : Factores que ponen en problema un proyecto. ....	15
Tabla 1.4 : Factores que causan una cancelación de un proyecto. ....	16
Tabla 1.5 : Factores que contribuyen al éxito de un proyecto. ....	16
Tabla 4.1 : Problemas Abordados vs. Actividades del Modelo de Proceso de la IR .....	64
Tabla 4.2 :Cuadro de Técnicas VS las diferentes actividades de la Ingeniería de Requerimientos .....	116
Tabla 4.3: Plantilla de Objetivos del Sistema .....	119
Tabla 4.4: Plantilla para requisitos de almacenamiento de información .....	120
Tabla 4.5: Plantilla para Requerimientos Funcionales .....	122
Tabla 4.6: Plantilla para Caso Uso .....	123
Tabla 4.7: Plantilla para requerimiento no funcional.....	125
Tabla 4.8: Plantilla para conflictos .....	126
Tabla 4.9: Plantilla para Clases .....	127
Tabla 4.10: ..... Plantilla de Asociación 128	
Tabla 4.11: Tabla de Listado de los objetivos del Sistema.....	131
Tabla 8.1: Análisis comparativo de algunas metodologías de obtención de requerimientos.....	173

## **INTRODUCCIÓN**

Actualmente los sistemas computacionales están presentes en todos los aspectos de la vida moderna. Basta por ejemplo mencionar sistemas de información en medicina, en la industria aeroespacial, etc. Un producto de software y la información que maneja constituyen activos fundamentales en las empresas que los poseen, cualquiera sea su naturaleza.

Es importante notar que los productos de software son útiles principalmente para obtener de la información almacenada, nueva información derivada que permita tomar decisiones y obrar. Por consiguiente, es natural suponer que de la confiabilidad del sistema de software se deducirá la confiabilidad de la información resultante. Los errores en el software pueden producir en un plazo no determinado perjuicios no sólo económicos, sino también pérdida de vidas humanas, daños ambientales de diversa gravedad, etc.

Desde el momento en que los requerimientos son necesidades que deben satisfacer los sistemas a ser construidos, estos conjuntos de requerimientos definirán el éxito o fracaso de los proyectos, entonces tiene sentido buscar lo que son los requerimientos, escribirlos, organizarlos y seguirlos en el momento en que cambian. De manera tal que se minimice el riesgo de que se produzcan defectos y aumente el esfuerzo de retrabajo. En síntesis, para mejorar la calidad de los productos.

Debido a esto se necesita entonces la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema. Se necesita una disciplina que establezca el proceso de definición de requerimientos como una sucesión de actividades mediante la cual lo que debe hacerse se "elicit", se modela y analiza. En este proceso se deben conciliar diferentes puntos de vista y utilizar una combinación de métodos, personas y herramientas. El resultado final constituirá la documentación de los requerimientos. Éstos deben expresarse de forma clara y estructurada de manera que puedan ser

entendidos tanto por expertos como por el usuario, quien deberá participar en la validación. A este concepto se le llama Ingeniería de Requerimientos.

De esta manera la Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental:

**la definición de lo que se desea producir.**

Por consiguiente el desarrollar una metodología de ingeniería de requerimientos es tener una alternativa para poder capturar requerimientos suficientemente buenos y de esta manera lograr que se pueda continuar con las fases de diseño y construcción con un adecuado nivel de calidad.

## 1. PLANTEAMIENTO DEL PROBLEMA

### 1.1 Descripción de la Realidad

Actualmente, se observa muchos fallos en los proyectos de software, hay muchos proyectos en la cual son víctimas de retrasos, presupuestos sobre girados y con problemas de calidad. A pesar de que un número creciente de herramientas automatizadas han surgido para ayudar a definir y aplicar un proceso de desarrollo de software efectivo, aún existen procesos de producción informal, parcial y en algunos casos no confiables.

Los errores cometidos al entender y definir los requerimientos tienen el potencial de ser los de mayor costo ya que muchas decisiones de diseño dependen de estos. Los requerimientos son la base fundamental del desarrollo de un sistema ya que especifica que es lo que el sistema debe hacer.

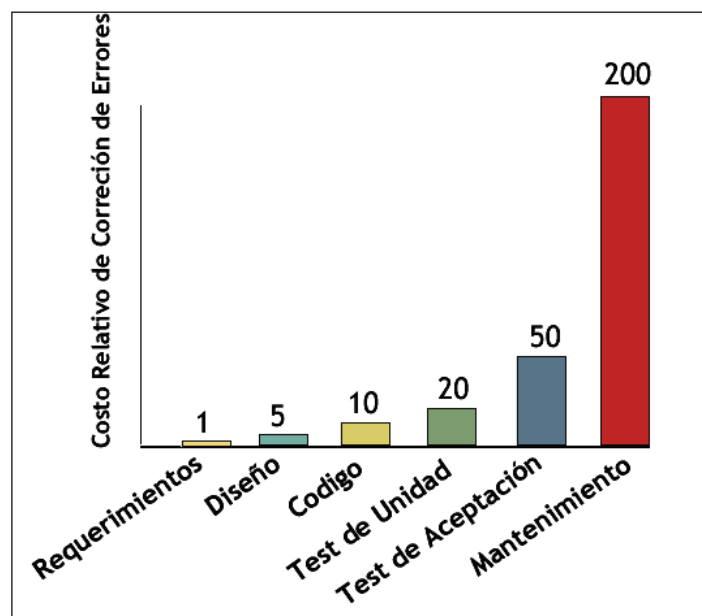


Figura 1.1 Costo de Corrección Tardía (Boehm, 1981)



Figura 1.2 : Problemas de Desarrollo: Un ejemplo regional(Chile)

	1994	1998	2003	2004
EXITOSOS	16%	26%	16%	29%
CON PROBLEMAS	53%	46%	54%	53%
CANCELADOS	31%	28%	30%	18%

Tabla 1.1 : Relevamiento de proyectos de software en EEUU por el Standish Group.



**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

	EXITOSO	CON PROBLEMAS	CANCELADO
1	USUARIOS INVOLUCRADOS	FALTA DE INPUT DE LOS USUARIOS	REQUERIMIENTOS INCMPLETOS
2	APOYO DE GERENCIA EJECUTIVA	REQUERIMIENTOS INCOMPLETOS	FALTA DE INPUT DE LOS USUARIOS
3	CLARA DESCRIPCIÓN DE REQUERIMIENTOS	REQUERIMIENTOS CAMBIANTES	FALTA DE RECURSOS

**Tabla 1.2** Causa percibida de éxito y fracaso de un proyecto.

1.	FALTA DE INPUT DE LOS USUARIOS	12.8%
2.	REQUERIMEINTOS Y ESPECIFICACIONES INCOMPLETAS	12.3%
3.	REQUERIMIENTOS Y ESPECIFICACIONES CAMBIANTES	11.8%
4.	FALTA DE APOYO DE GERENCIA EJECUTIVA	7.5%
5.	INCOPIENCIA TECNICA	7.0%
6.	FALTA DE RECURSOS	6.4%
7.	EXPECTATIVAS NO RELISTAS	5.9%
8.	OBJETIVOS POCO CLAROS	5.3%
9.	TIEMPOS POCO REALISTAS	4.3%
10.	TECNOLOGÍA NUEVA	3.7 %
	OTROS	23.0%

**Tabla 1.3 :** Factores que ponen en problema un proyecto.

FACTORES QUE CAUSAN LA CANCELACIÓN DEL PROYECTO		
1	REQUERIMIENTOS INCOMPLETOS	13.1%
2	FALTA DE INVOLUCRAMIENTO DE USUARIOS	12.4%
3	FALTA DE RECURSOS	10.6%
4	EXPECTATIVAS NO REALISTAS	9.9%
5	FALTA DE APOYO GERENCIAL	9.3%
6	REQUERIMIENTOS Y ESPECIFICACIONES CAMBIANTES	8.7%
7	FALTA DE PLANIFICACION	8.1%
8	“YA NO LO NECESITAMOS”	7.5%
9	FALTA DE GESTIÓN	6.2%
10	INCOMPETENCIA TECNOLÓGICA	4.3%
	<b>OTROS</b>	<b>9.9%</b>

**Tabla 1.4 : Factores que causan una cancelación de un proyecto.**

FACTORES QUE CONTRIBUYEN AL ÉXITO DE UN PROYECTO		
1.	INVOLUCRAMIENTO DE USUARIOS	15.9%
2.	APOYO DE GERENCIA EJECUTIVA	13.9%
3.	DESCRIPCION DE REQUERIMIENTOS CLARA	13.0%
4.	PLANIFICACION APROPIADA	9.6%
5.	EXPECTATIVAS REALISTAS	8.2%
6.	ENTREGAS	7.7%
7.	PERSONAL COMPETENTE	7.2%
8.	OWNERSHIP	5.3%
9.	VISION Y OBJETOS CLAROS	2.9%
10.	<b>OTROS</b>	<b>13.9%</b>

**Tabla 1.5 : Factores que contribuyen al éxito de un proyecto.**

Estos cuadros muestran los puntos en lo que muchos proyectos de software fracasan por no realizar un estudio correcto de los requerimientos.

Esto indica que los principales problemas que han dado origen a la crisis del software residen en las primeras etapas del desarrollo, cuando hay que decidir las características del producto software a desarrollar.

Uno de los hechos también comprobados es la demora de tiempo y el costo de un cambio en los requerimientos, una vez entregado el producto, es entre 60 y 100 veces superior al que hubiera representado el mismo cambio durante las fases iniciales de desarrollo, por lo que no es de extrañar que aquellos proyectos en los que no se determinan correctamente los requerimientos y cambian frecuentemente durante el desarrollo, superen con creces el tiempo y presupuesto inicial.

Analizando las diferentes causas que dan lugar a un software desarrollado insatisfactoriamente, se observó los siguientes aspectos negativos:

- Pocas herramientas de soporte tecnológico, para realizar la documentación de requerimientos de software aumentando el tiempo y costos iniciales del proyecto.
- En la etapa de adquisición de requerimientos frecuentemente existe contradicciones y ambigüedad que atentan contra el correcto comienzo de la vida del software.
- Existe situaciones en que es escaso el conocimiento sobre el dominio de aplicación.
- El dominio de aplicación, donde se desarrollará el software, puede ser complejo.

## **1.2 Antecedentes y Formulación del Problema**

A principios de los años 50 en algún organismo del gobierno de los Estados Unidos se podía encontrar una sala de un centro de cálculo.

La estancia estaba ocupada por un enorme ordenador cuyo desarrollo había costado varios millones de dólares y cuyo mantenimiento tenía ocupadas a varias personas las 24 horas del día.

La siguiente media hora era el turno semanal de uno de los muchos científicos que trabajaba para el Departamento de Defensa. Su trabajo consistía en calcular tablas numéricas para trayectorias balísticas usando ecuaciones relativamente complejas. Llevaba toda la semana repasando su programa en su despacho para asegurarse que no contenga errores sintácticos ni semánticos, ya que un fallo en la compilación podría hacerle perder su preciosa media hora semanal.

Cuando comienza su tiempo, el ordenador comienza a leer las tarjetas perforadas que previamente había entregado al operador, compila el programa sin errores y comienza su ejecución. El científico se dispone a esperar a que la impresora comience a imprimir las tablas. Cuando faltan sólo pocas líneas para completarlas, el hardware falla y los encargados de mantenimiento deben parar la máquina para repararla. Quizás la semana que viene tenga más suerte.

Si se compara esta situación con la actual, pocos son los aspectos comunes que se encuentran. Sin embargo, se podría hacer un ejercicio de análisis e identificar algunas de las características del desarrollo de software en los comienzos de la informática.

- ***El hardware era mucho más caro que el software.*** La máquina y su mantenimiento costaban millones de dólares. Comparado con este costo, el sueldo del científico que escribe el programa era ridículo, así que ***¿por qué preocuparse por el costo del desarrollo de software?***
  
- ***El desarrollo del hardware era más complejo que el del software.*** La tecnología hacía que el hardware sea complejo de construir y mantener. El software habitual solía ser programas no muy grandes (debido, entre otras limitaciones, a la escasa capacidad de memoria) y estaban escritos por una única persona, normalmente empleada en la organización que utiliza el hardware. Los requerimientos que tenía que cumplir el software eran simples. Por lo tanto, ***¿por qué preocuparse por la complejidad del software?***

- 
- ***El hardware era poco fiable.*** Debido a la tecnología que se utilizaba para su implementación, en cualquier momento la máquina podía sufrir una avería, así que ***¿para qué preocuparse por la calidad del software?***

Esta despreocupada situación respecto al software cambia cuando, gracias a los avances en la tecnología, aumenta la capacidad de memoria y se reducen los costos de desarrollo y mantenimiento del hardware.

Se empiezan a comercializar los primeros ordenadores y la demanda de software más complejo crece rápidamente, generando la crisis del software, término utilizado por primera vez en la conferencia organizada por la Comisión de Ciencia de la OTAN en Garmisch, Alemania, en octubre de 1968, para designar la gran cantidad de problemas que presentaba, y aún presenta, el desarrollo de software y el alto índice de fracasos en los proyectos de desarrollo.

***¿Qué podía hacerse ante una situación en la que los proyectos software tenían un alto riesgo de fracasar?*** La respuesta parecía obvia: construir software de forma similar a como se construye hardware, aviones, barcos, puentes o edificios, es decir, aplicar los métodos de la ingeniería a la construcción de software.

Desde 1968 se ha invertido un gran esfuerzo en determinar las causas y proponer soluciones para la crisis del software.

En 1979, la Oficina de Cuentas del Gobierno Norteamericano (*Government Account Office*, GAO) realizó un estudio [GAO, 1979] seleccionando nueve proyectos de desarrollo de software para el gobierno norteamericano cuyos contratos sumaban una cantidad total de 6.800.000 dólares.

De esta cantidad, sólo 119.000 dólares correspondían a un proyecto que se había utilizado tal como se había entregado. Dicho proyecto se trataba de un preprocesador de COBOL, por lo que era un problema relativamente simple cuyos requerimientos **eran comprendidos por clientes y desarrolladores y que además no cambiaron durante el desarrollo.**

El resto de los 6.8 millones de dólares se distribuyeron como puede verse en la figura 1.3, en la que puede destacarse el enorme porcentaje de dinero invertido en proyectos cancelados o no satisfactorios.

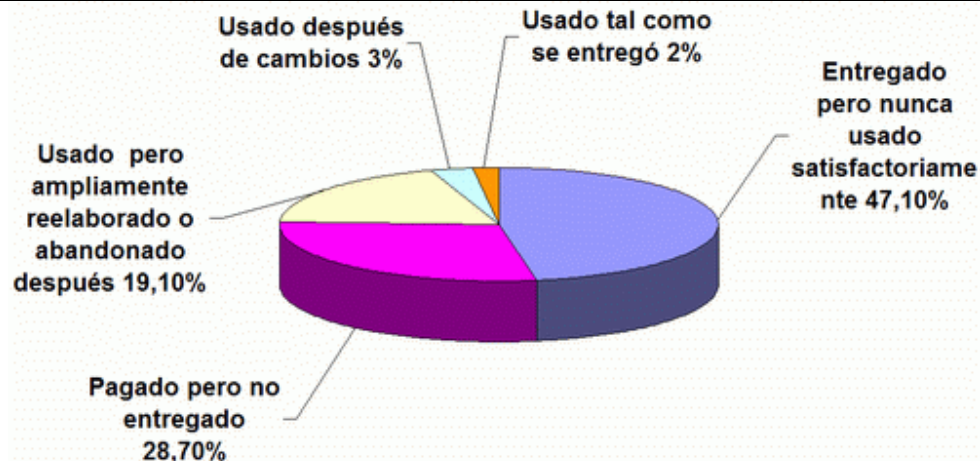


Figura 1.3: Resultados del Informe de GAO.

Fuente: Government Account Office, GAO [GAO, 1979].

En 1995, el Grupo Standish realizó un estudio, el informe CHAOS, mucho más amplio y significativo que el del GAO cuyos resultados, a pesar de haber pasado más de 25 años, no reflejaban una mejoría sustancial [TSG, 1995].

Los resultados generales, que pueden verse en la figura 1.4, si se compara con los de [GAO, 1979] presentan una mejora en los proyectos que se entregan cumpliendo todos sus requerimientos, 2% frente al 16.2%, sólo el 9% en grandes compañías, pero empeoran ligeramente respecto a los que se abandonan durante el desarrollo, 28.7% frente a 31.1%.

Sin incluir al 16.2% de los proyectos terminados correctamente, **la media del gasto final fue del 189% del presupuesto original, el tiempo necesario para su realización del 222% del plazo original y se cumplieron una media del 61% de los requerimientos iniciales**, cifras que también empeoraban en el caso de grandes compañías.

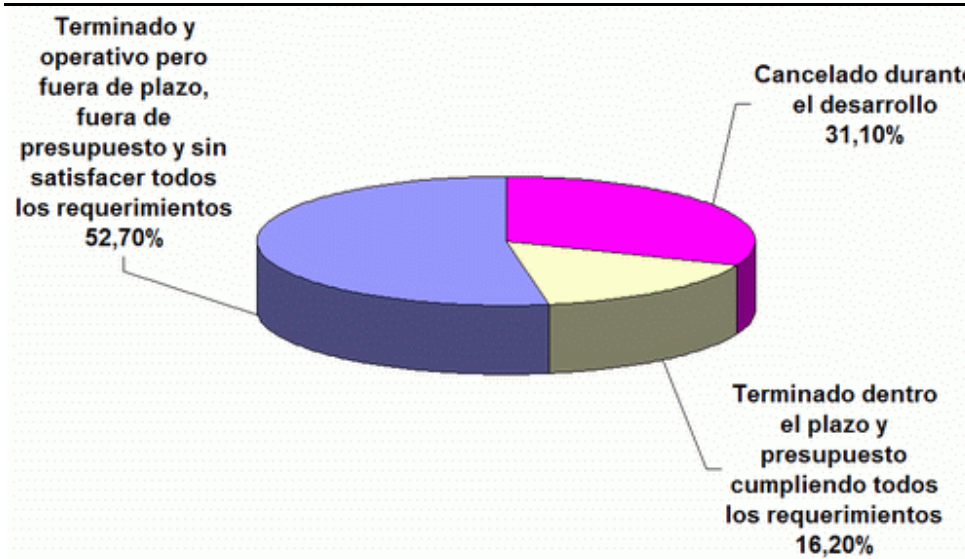


Figura 1.4: Resultados del informe CHAOS.

Fuente: Grupo Standish [TSG 1995].

En 1996, el proyecto ESPITI (*European Software Process Improvement Training Initiative*) [ESP, 1996] realizó una investigación sobre los principales problemas en el desarrollo de software a nivel europeo. Los resultados, muy similares a los obtenidos en el informe CHAOS, indicaron que los mayores problemas estaban también relacionados con la especificación, la gestión y la documentación de los requerimientos de software.

Todas estas circunstancias han convencido a la gran parte de la comunidad de la ingeniería del software de la necesidad, cada vez mayor, de una **ingeniería de requerimientos**.

### 1.3 Importancia y Justificación

- **La importancia del proyecto** es que se define una metodología que permitirá la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema. De esta manera se proporciona una herramienta de apoyo para obtener, analizar y especificar los requerimientos constituyéndose importante

ya que ayudará para los desarrolladores, analistas, ingenieros de software, ingenieros de requerimientos, líderes del proyecto, etc., a entender lo que realmente se obtendrá del sistema a desarrollar.

- Tiene una **importancia económica** ya que disminuye los costos y retrasos del proyecto: Muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la Ingeniería de Requerimientos. Errores cometidos al entender los requerimientos tienen el potencial de ser los de mayor costo, porque muchas decisiones de diseño dependen de estos.
- **Mejora la capacidad de predecir cronogramas de proyectos**, así como sus resultados, ya que ayudará a obtener requerimientos completos, consistentes y no ambiguos, proporcionando un punto de partida para estimación de costos, tiempo y recursos necesarios.
- **Permite gestionar las necesidades del proyecto en forma estructurada y ordenada:** Cada actividad de la metodología (elicitación, análisis, especificación y validación) consiste de una serie de pasos organizados, bien definidos.
- **Mejora la calidad del software:** La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, portabilidad, mantenibilidad, etc.).
- **Mejora la comunicación entre equipos:** La especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
- **Evita rechazos de usuarios finales:** La ingeniería de requerimientos obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto.



La justificación para escoger este tema se fundamentó en la gran cantidad de proyectos de software que no llegan a cumplir sus objetivos o que fracasan. Estudios realizados muestran que más del 53% de los proyectos de software fracasan por no realizar un estudio previo de requerimientos. Otros factores como falta de participación del usuario, requerimientos incompletos y el cambio a los requerimientos, también ocupan sitios altos en los motivos de fracasos. Otro dato interesante revela que el costo de solucionar un error en la etapa de mantenimiento es aproximadamente 200 veces mayor que solucionarlo en la etapa de requerimientos. Por todo esto al desarrollar una **Metodología para la Ingeniería de Requerimientos** demostraremos como la calidad del producto de Software depende de seleccionar las mejores técnicas, utilizando plantillas, formatos, diagramas para una efectiva captura de los requerimientos.

#### **1.4 Definición del Problema**

Para comenzar el planteamiento del problema, es conveniente citar a Frederick Brooks:

*La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan dificultosa como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros sistemas software. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si es realizada en forma errónea. Ninguna otra parte es tan dificultosa de rectificar posteriormente.* (Frederick P. Brooks, Jr., The Mythical Man-Month, Addison-Wesley, 1995).

La adquisición de requerimientos es una de las más importantes partes del proceso de desarrollo de software, pero es a la vez una de las que cuentan con menor soporte en la actualidad. A su vez es una etapa donde inevitablemente existe ambigüedad, incompletitud, contradicciones que atentan contra el correcto comienzo de la vida del producto. (Brooks, 1995).

Las causas primarias de realizar una incorrecta conceptualización del problema pueden clasificarse de la siguiente manera: (Sommerville, 1996)

- ✓ Carencia de conocimientos sobre el dominio
- ✓ Dominio de aplicación complejo
- ✓ Carencia de experiencia en detección de requerimientos, conceptualización de los mismos, analogía con sistemas anteriores, etc.

Tales problemas sugieren que existe una creciente necesidad de herramientas o técnicas que asistan en el proceso de elaborar los requerimientos de un sistema software, que trabaje con dominios complejos de modo que colabore con los analistas en dicha tarea.

**Una metodología de ingeniería de requerimientos es una importante ayuda para poder definir clara y consistentemente los requerimientos del sistema a desarrollar.**

Una herramienta que contribuya en el mejoramiento de la práctica de la ingeniería del software y que a la vez incentive su aplicación sería de vital importancia.

### **1.5 Limitaciones y Alcances**

El alcance de la investigación es definir herramientas y las técnicas necesarias para las principales actividades de la ingeniería de requerimientos, desarrollando una metodología que defina las tareas recomendadas para cada actividad de la IR, después de planteada la metodología se aplicara en el escenario propuesto: Sistema Logístico de una Empresa Distribuidora de Alimentos.

Nuestra Investigación tendrá como producto final una metodología; así como un caso Practico de esta, entregaremos un documento de requerimientos que contenga cada tarea con las técnicas planteadas.

Debido al Tamaño del Sistema Logístico solo se analizará los requerimientos de sus principales tareas.

## **1.6 Variantes del Problema**

La metodología Propuesta puede ser extensible a cualquier Sistema aunque no necesariamente se deba utilizar todas las tareas propuestas pues cada organización es diferente y particular.

Cabe resaltar que no existe un proceso único que sea válido de aplicar en todas las organizaciones. Cada organización debe desarrollar su propio proceso de acuerdo al tipo de producto que se esté desarrollando, a la cultura organizacional, y al nivel de experiencia y habilidad de las personas involucradas en la ingeniería de requerimientos. Hay muchas maneras de organizar el proceso de ingeniería de requerimientos y muchas veces tenemos también que recurrir a consultores, ya que ellos tienen una perspectiva mas objetiva que las personas involucradas en el proceso.

## **2. OBJETIVOS**

### **2.1 Objetivo General**

El objetivo de este trabajo es definir una metodología para la elicitación y análisis de requerimientos que pueda ser utilizado como guía para determinar, analizar y especificar los requerimientos de un proyecto en el cual el dominio del problema es totalmente desconocido y no es fuertemente estructurado. En resumen, el trabajo pretende mostrar las herramientas existentes que se pueden utilizar en cada tarea identificada para la metodología y de que forma se usaran (técnica) en el transcurso del proceso de Ingeniería de requerimientos.

### **2.2 Objetivo Específicos**

- Definir las actividades para la obtención de requerimientos.
- Definir de las tareas a realizar.
- Describir las técnicas y herramientas en las actividades de elicitación, análisis y especificación de requerimientos.
- Definir plantillas para la descripción de requerimientos.
- Definir los modelos del sistema usando UML 2.0.
- Definir la estructura del documento de requerimientos, que es el producto final de la metodología propuesta.

### **3. MARCO TEORICO CONCEPTUAL**

#### **3.1 Antecedentes de la investigación**

Se han venido realizando en el extranjero, estudios y aplicaciones para la implementación de una metodología.

- “UNA GUÍA PARA EXTRAER, ANALIZAR, ESPECIFICAR Y VALIDAR LOS REQUERIMIENTOS DE UN PROYECTO”  
Proyecto desarrollado Nicolás Davyt Dávila, Uruguay
- “AMIR-ST: PROPUESTA DE UNA APROXIMACIÓN METODOLÓGICA PARA LA INGENIERÍA DE REQUERIMIENTOS DE SISTEMAS TELEMATICOS”  
Proyecto desarrollado por Mario Fernando Solarte Sarasty. España.
- “ESTIMACIÓN DE COMPLETITUD EN MODELOS DE REQUERIMIENTOS BASADOS EN LENGUAJE NATURAL”  
Proyecto desarrollado por Marcela Ridao INTIA Facultad de Ciencias ExactaUNICEN, Tandil, Argentina

## 3.2 Bases Teóricas

### 3.2.1 La Ingeniería de Software

Según la definición del IEEE, citada por [Lewis 1994] "*software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo*". Según el mismo autor, "*un producto de software es un producto diseñado para un usuario*". En este contexto, la ingeniería de software (SE del inglés *Software Engineering*) es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software", que en palabras más llanas, se considera que la ingeniería de software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos" [Cota 1994].

### 3.2.2 El proceso de ingeniería de software

Se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad" [Jacobson 1998].

### 3.2.3 El proceso de desarrollo de software

"Es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" [Jacobson 1998].

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

Como vemos la necesidad de una ingeniería del software es clara para poder llevar a cabo un Proyecto de Software de calidad. Sin embargo, la ingeniería del software comprende una serie de actividades lo suficientemente diversas como para poder considerar la necesidad de otras ingenierías dentro de la propia ingeniería del software: Ingeniería de Requerimientos.

### **3.2.4 La necesidad de la Ingeniería de Requerimientos**

Ya se han convencido a la gran parte de la comunidad de la ingeniería del software de la necesidad, cada vez mayor, de una ingeniería de requerimientos.

Ejemplos de este convencimiento son, entre otros:

1. El hecho de que el Software Engineering Institute de la Universidad Carnegie–Mellon haya identificado la gestión de requerimientos como una de las áreas de proceso clave (key process area) dentro del nivel 2 (Repetible) del Capability Maturity Model (CMM) [Paulk et al. 1993].
2. La organización de congresos específicos como:
  - El IEEE International Symposium on Requirements Engineering (RE), que se celebra los años impares desde 1993 organizado por IEEE, ACM, IFIP y varias asociaciones más.
  - La IEEE International Conference on Requirements Engineering (ICRE), que se celebra los años pares desde 1994 organizado por el IEEE.
  - El Workshop em Engenharia de Requerimientos (WER), que se celebra anualmente desde 1998.
3. La publicación de revistas especializadas como el Requirements Engineering Journal, que se publica trimestralmente desde 1996.
4. La aparición bianual de números monográficos sobre ingeniería de requerimientos en IEEE Software coincidiendo con la celebración del ICRE, en concreto los correspondientes a los meses de marzo de los años 1994, 1996 y 1998 y el del mes de mayo del año 2000.
5. La financiación pública de proyectos europeos como:
  - NATURE (Novel Approaches to Theories Underlying Requirements Engineering).

- REAIMS (Requirements Engineering Adaptation and IMprovement for Safety and dependability).
- CREWS (Cooperative Requirements Engineering With Scenarios)
- La financiación de proyectos nacionales como el proyecto CICYT MENHIR (Metodologías, Entornos y Nuevas Herramientas para la Ingeniería de Requerimientos)
- La red europea RENOIR (Requirements Engineering Network Of International cooperating Research groups)

Una vez asumida la necesidad de una ingeniería de requerimientos, el reto al que se enfrenta la comunidad investigadora es identificar claramente los problemas a los que se enfrenta y dotar a dicha ingeniería de las herramientas teóricas y prácticas necesarias para solucionarlos, teniendo en cuenta la dificultad añadida de lo costoso de la obtención de datos empíricos tanto en ingeniería del software en general, como en ingeniería de requerimientos en particular.

La motivación del trabajo que se describe en esta tesina es realizar aportaciones a la ingeniería de requerimientos, especialmente en los aspectos metodológicos y de herramientas que soporten el proceso. Dentro de las posibles clases de sistemas informáticos (sistemas de información, sistemas empujados y sistemas de control [Sawyer y Kontoya 1999]), este trabajo se centra en la ingeniería de requerimientos de sistemas de información.

### **3.2.5 Ingeniería de Requerimientos**

La Ingeniería de Requerimientos se define, según Ortas [Ortas 1997], como un "conjunto de actividades en las cuales, utilizando técnicas y herramientas, se analiza un problema y se concluye con la especificación de una solución (a veces más de una)."

Algunos otros conceptos de ingeniería de requerimientos son:

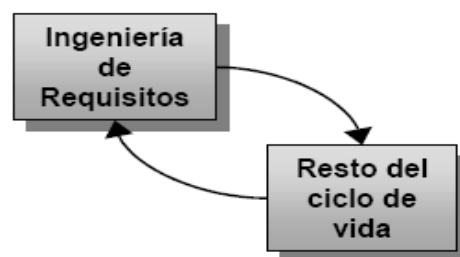
"Ingeniería de Requerimientos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software". (Pressman, 2006: 155)



“La ingeniería de requerimientos es el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema”. (Sommerville, 2005: 82)

Entonces, "Ingeniería de Requerimientos" se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requerimientos para un producto determinado. El uso del término "ingeniería" implica que se deben utilizar técnicas sistemáticas y repetibles para asegurar que los requerimientos del sistema estén completos y sean consistentes y relevantes.

En nuestra opinión, la ingeniería de requerimientos puede considerarse como un proceso de descubrimiento y comunicación de las necesidades de clientes y usuarios y la gestión de los cambios en dichas necesidades. Su situación dentro del ciclo de vida de desarrollo de software puede verse en la figura 1.3, aunque debe tenerse en cuenta que la ingeniería de requerimientos continúa durante todo el proceso de desarrollo [Sawyer y Kontoya 1999].



**Figura 3.1 La ingeniería de requerimientos en el ciclo de vida del desarrollo de Software**

Desde nuestro punto de vista, el aspecto más importante en la ingeniería de requerimientos es la comunicación, como se discutirá más adelante. Esta característica es la que hace de la ingeniería de requerimientos una disciplina especialmente compleja ya que hay un factor que, aunque lleva siendo estudiado mucho tiempo, apenas se conoce aún: el factor humano.

Este factor es el responsable de que la ingeniería de requerimientos tenga aspectos sociales y culturales y no sólo técnicos [Goguen 1994].



**Figura 3.2: La Ingeniería de requerimientos como un proceso de Comunicación**

### **3.2.6 Importancia de la ingeniería de requerimientos**

Según la autora Lizka Johany Herrera en su documento de la ingeniería de requerimientos, los principales beneficios que se obtienen de la Ingeniería de Requerimientos son (2003: 3):

1. Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad de la IR consiste de una serie de pasos organizados y bien definidos.
2. Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
3. Disminuye los costos y retrasos del proyecto: es sabido que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la IR, ya que es una de las etapas de mayor importancia en el ciclo de desarrollo de software y de las primeras en llevarse a cabo.
4. Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).

### **3.2.7 Requerimientos**

Se presenta a continuación la definición existente en el glosario de la IEEE de lo que es un “Requerimiento”:

1. “Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”. (Std 610.12-1900, IEEE: 62)
2. “Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal”. (Std 610.12-1900, IEEE: 62)

También, Ian Sommerville presenta una definición acerca de lo que es un “Requerimiento”:

3. “Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste”. (Sommerville, 2005: 108)

Analizando las definiciones anteriores, un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso.

#### **3.2.7.1 Tipos de Requerimientos**

- **Requerimientos que definen efectos sobre el entorno**  
“El sistema mantendrá la temperatura de la caldera entre 70° y 80°.”
- **Requerimientos muy generales**  
“El sistema mantendrá un registro de todos los materiales de la biblioteca, incluyendo libros, periódicos, revistas, videos y CD -Rooms.”

- **Requerimientos Funcionales**

Describen los servicios (funciones) que se esperan del sistema. Ejemplo: “El sistema aceptará pagos con VISA”

- **Requerimientos No Funcionales**

Restricciones sobre los requerimientos funcionales. Ejemplo:

“El sistema aceptará pagos con VISA de forma segura y con un tiempo de respuesta menor de 5 segundos”.

Se distinguen dos tipos de requerimientos no funcionales:

1. **Orientados al usuario:**

Fiabilidad, Seguridad (*security*), Seguridad (*safety*), Usabilidad, Robustez, Disponibilidad, Rendimiento (Tiempo de respuesta, Capacidad, *Throughput*).

2. **Orientados al desarrollador:**

Disponibilidad, Portabilidad, Adaptabilidad, Testabilidad, Comprensibilidad.

Pero esta distinción, muchas veces, resulta arbitraria: Ejemplo:

“El sistema aceptará pagos con VISA a través del protocolo SET”

- **Requerimientos de implementación**

“El interfaz de usuario se implementará sobre un navegador Web.”

- **Requerimientos de rendimiento**

“El sistema deberá soportar al menos 20 transacciones por segundo.”

- **Requerimientos de usabilidad:**

“El sistema permitirá que los nuevos usuario se familiaricen con su uso en menos de 15 minutos.”

- **Requerimientos en negativo**

Es importante decir lo que el sistema no debe hacer, empleando requerimientos “en negativo” que limiten el ámbito del sistema. Indican dónde no se deben emplear recursos. Fundamental para sistemas críticos. Se debe mantener la distinción liveness/safety del conjunto de requerimientos:

**Liveness:** Requerimientos que dicen lo que el sistema debe hacer.

**Safety:** Requerimientos que dicen lo que el sistema no debe hacer.

- **Requerimientos de comunicaciones del sistema**

Son requerimientos de carácter técnico relativos a las comunicaciones que deberá soportar el sistema software a desarrollar. Por ejemplo: el sistema deberá utilizar el protocolo TCP/IP para las comunicaciones con otros sistemas.

- **Requerimientos de interfaz de usuario**

Este tipo de requerimientos especifica las características que deberá tener el sistema en su comunicación con el usuario. Por ejemplo: la interfaz de usuario del sistema deberá ser consistente con los estándares definidos en IBM's Common User Access.

Se debe ser cuidadoso con este tipo de requerimientos, ya que en esta fase de desarrollo todavía no se conocen bien las dificultades que pueden surgir a la hora de diseñar e implementar las interfaces, por esto no es conveniente entrar en detalles demasiado específicos.

- **Requerimientos de fiabilidad**

Los requerimientos de fiabilidad deben establecer los factores que se requieren para la fiabilidad del software en tiempo de explotación. La Fiabilidad mide la probabilidad del sistema de producir una respuesta satisfactoria a las demandas del usuario. Por ejemplo: la tasa de fallos del sistema no podrá ser superior a 2 fallos por semana.

- **Requerimientos de entorno de desarrollo**

Este tipo de requerimientos especifican si el sistema debe desarrollarse con un producto específico. Por ejemplo: el sistema deberá desarrollarse con Oracle 7 como servidor y clientes Visual Basic 6.

- **Requerimientos de portabilidad**

Los requerimientos de portabilidad definen qué características deberá tener el software para que sea fácil utilizarlo en otra máquina o bajo otro sistema operativo. Por ejemplo: el sistema deberá funcionar en los sistemas operativos Windows 95, Windows 98 y Windows NT 4.0, siendo además posible el acceso al sistema a través de Internet usando cualquier navegador compatible con HTML 3.0.

Debido a que hay tantos tipos distintos de requerimientos, no es posible establecer una forma estándar de escribirlos. Tampoco es posible decir cual es “la mejor forma” de especificarlos. Todo depende mucho de quien los escribe, quien los va a leer, el dominio de la aplicación, etc.

### **¿Relación Requerimientos-Arquitectura?**

La elección de una determinada arquitectura software debe tener en cuenta los requerimientos funcionales pero, sobre todo, los requerimientos no funcionales. No hay una regla definitiva para establecer, dados los requerimientos, el tipo de arquitectura. Tan sólo hay una serie de heurísticas para, dados unos requerimientos, elegir la arquitectura.

#### **3.2.7.2 Dificultades para definir los requerimientos**

Durante la etapa de especificación de requerimientos se pueden presentar muchos inconvenientes los cuales son importantes de identificar y prevenir, a continuación se presenta un listado con los problemas más comunes en este proceso:

1. Los requerimientos no son obvios y vienen de muchas fuentes.
2. Son difíciles de expresar en palabras (el lenguaje es ambiguo).
3. La cantidad de requerimientos en un proyecto puede ser difícil de manejar.

4. Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.
5. El usuario no puede explicar lo que hace
6. Tiende a recordar lo excepcional y olvidar lo rutinario
7. Hablan de lo que no funciona
8. Los usuarios tienen distinto vocabulario que los desarrolladores.
9. Usan el mismo término con distinto significado

### **3.2.7.3 Propiedades deseables de los requerimientos**

Sobre las propiedades que deben tener los requerimientos, o más concretamente las especificaciones de requerimientos, hay una amplia bibliografía. A continuación enumeramos las propiedades que consideramos más importantes después de analizar las consideradas, entre otros, en [IEEE 1996], [IEEE 1993], [Davis 1993] y [Wieringa 1996].

1. **Comprensible por clientes y usuarios:** la ingeniería de requerimientos es básicamente un problema de descubrimiento y comunicación de las necesidades de clientes y usuarios, por lo que la propiedad más importante de una especificación es que sirva como canal de comunicación entre los stakeholder en el proceso de ingeniería de requerimientos (ver figura 3.3). Evidentemente, si una especificación de requerimientos debe servir de canal de comunicación, es imprescindible que sea comprensible por parte de los clientes y usuarios. En nuestra opinión, la mejor forma de lograr esta comunicación es pensar en la audiencia a la que van dirigidos los requerimientos, de forma que para comunicarse con los clientes y usuarios lo mejor es utilizar requerimientos–C y para hacerlo con los desarrolladores lo mejor es usar requerimientos–D (ver figura 3.3).

En [Wieringa 1996] se considera esta propiedad como una característica, junto con la no ambigüedad, de la propiedad más importante que debe tener una especificación de requerimientos: la comunicabilidad.

En [Davis 1993] y [Davis 1995] se advierte de un uso indiscriminado de notaciones formales que hagan imposible entender la especificación a los clientes y usuarios, que normalmente no tienen la formación necesaria para entender este tipo de notaciones.

En [IEEE 1996] se hace especial hincapié en que la especificación de requerimientos debe hacer de puente entre los clientes y los desarrolladores, y que por lo tanto debe ser comprensible para ambos grupos.

En [IEEE 1993, pág. 9] se advierte de la necesidad de que la especificación de requerimientos sea preparada conjuntamente por clientes, usuarios y desarrolladores, aunque no especifica ningún estilo para su redacción.

2. **Correcta:** una especificación de requerimientos es correcta si y sólo si todo requisito contenido en ella representa alguna propiedad requerida por el sistema a desarrollar [IEEE 1993, Davis 1993]. En otras palabras, ningún requisito debe ser innecesario. Deben formar un conjunto único, siguiendo la nomenclatura de [IEEE 1996]. Interpretando los requerimientos como restricciones, añadir requerimientos innecesarios implicaría reducir innecesariamente el espacio de sistemas válidos (ver figura 1.8, en la que el requisito R4 es un requisito innecesario). Este espacio reducido de sistemas válidos  $S_{0v}$ , sería un subconjunto del verdadero espacio de sistemas **válidos**  $S_v$ .

3. **No ambigua:** una especificación de requerimientos no es ambigua si y sólo si todo requisito contenido en ella tiene una sola interpretación [IEEE 1993, Davis 1993]. Para evitar interpretaciones erróneas conviene que aquellos términos con más de una posible interpretación aparezcan en un glosario de términos en el que se especifique claramente su significado en el documento. En el caso de que se utilice lenguaje natural para especificar los requerimientos, que es lo habitual al especificar los requerimientos—C, se ha de tener especial cuidado con esta propiedad.

En [Wieringa 1996], la no ambigüedad se considera como una característica de la comunicabilidad que debe tener la especificación de requerimientos.

4. **Completa:** una especificación de requerimientos es completa si cumple las siguientes propiedades [IEEE 1993, Davis 1993]:
  - a. Todo lo que se suponga que deba hacer el sistema a desarrollar está incluido en la especificación, es decir, no faltan requerimientos.



- b. Todas las respuestas del sistema a entradas tanto válidas como inválidas están especificadas, es decir, está especificada toda posible conducta del sistema desde un punto de vista externo o de caja negra.
- c. Todas las páginas, figuras y tablas están numeradas, todas las unidades de medida están definidas, todas las referencias externas son comprobables y no hay elementos por determinar (TBD, to be determined). En otras palabras, la especificación está organizada, los requerimientos son fáciles de localizar y el documento es sintácticamente correcto. En el caso de que haya elementos por determinar, es necesario conocer la causa de su no determinación y quién es responsable de su solución.

En [IEEE 1996], se considera que una especificación es completa si incluye todos los requerimientos identificados por el cliente y todos los requerimientos necesarios para la definición del sistema. En [Wieringa 1996], se considera una especificación como completa si no se omite ningún requisito. Desde un punto de vista práctico, esto significa que el cliente ha validado los requerimientos, es decir, que está de acuerdo con ellos y que confirma que no conoce más requerimientos que sean necesarios. En [Wieringa 1996] también se considera que los requerimientos deben estar priorizados por el cliente para considerar una especificación como completa.

Considerando los requerimientos como restricciones, si no se especifican todos los requerimientos se está definiendo un espacio incorrecto de sistemas válidos  $S_v$  0 que incluiría sistemas no válidos. Esta situación incrementa el riesgo de acabar desarrollando sistemas que no satisfagan las necesidades reales de clientes y usuarios.

5. **Consistente:** una especificación de requerimientos es consistente externamente si y sólo si todo requisito contenido en ella no está en conflicto con otros documentos de nivel superior [Davis 1993]. Es consistente internamente si y sólo si no existen conflictos entre los requerimientos que contiene. Según [IEEE 1993] y [Davis 1993], los conflictos entre requerimientos pueden ser de los siguientes tipos:

- Conflictos de conducta: dos o más requerimientos especifican conductas distintas del sistema para las mismas condiciones y el mismo estímulo externo.
- Conflictos de términos: se utilizan términos distintos para referirse al mismo concepto.
- Conflictos de característica: dos o más requerimientos especifican aspectos contradictorios para la misma característica del sistema.
- Conflictos temporales: dos o más requerimientos exigen características temporales contradictorias al sistema. En [IEEE 1996] también se comenta la necesidad de un mismo nivel de detalle y de un mismo estilo de redacción y de presentación de los requerimientos para considerar una especificación como internamente consistente.

Otra propiedad relacionada con la consistencia interna que se comenta en [IEEE 1996] es la necesidad de que la especificación esté limitada (bounded), es decir, que el ámbito y el contexto en el que se definen los requerimientos esté claramente identificado.

En [Wieringa 1996] se interpreta el término consistente como la propiedad de que pueda existir un sistema que satisfaga todos los requerimientos.

Evidentemente, si los requerimientos son contradictorios es imposible que pueda existir un sistema que los satisfaga todos a la vez.

6. **Verificable:** una especificación de requerimientos es verificable si y sólo si todo requisito contenido en ella es verificable, es decir, existe un proceso finito y de coste razonable por el que una persona o una máquina pueda comprobar que el sistema final cumple el requisito [Davis 1993, IEEE 1993]. Una condición absolutamente necesaria para que un requisito sea verificable es que no sea ambiguo y que se defina de forma medible, ya que no se puede comprobar algo que no se puede medir ni definir de forma precisa. Otra condición es que el proceso de verificación sea computable en el sentido de Turing. Por ejemplo, un requisito que indique que el sistema nunca podrá entrar en un bucle infinito no puede verificarse porque no existe ningún proceso que sea capaz de

hacerlo [Davis 1993]. Los procedimientos de observación para comprobar que el sistema cumple los requerimientos son la base para las pruebas de aceptación del sistema por parte del cliente [Wieringa 1996]. Considerando los requerimientos como restricciones, la verificabilidad y la no ambigüedad son necesarias para saber si un posible sistemas perteneciente al espacio de los sistemas posibles  $S_p$  cumple o no un determinado requisito  $R_i$ , es decir, si  $s$  pertenece a  $S_i$  o a  $\bar{S}_i$ .

7. **Modificable:** una especificación de requerimientos es modificable si y sólo si su estructura y estilo de redacción permite que los cambios se puedan realizar fácil, completa y consistentemente [Davis 1993, IEEE 1993]. Para conseguir esta propiedad, la especificación debe estar organizada coherentemente y debe contar con los índices y las tablas de referencias cruzadas oportunas, no debe ser redundante (debe estar normalizada según la terminología utilizada en [IEEE 1996]) y los requerimientos deben expresarse individualmente y no de forma conjunta. En el caso de que se opte por introducir redundancias para facilitar la legibilidad, dichas redundancias deben gestionarse con sumo cuidado, utilizando si es posible herramientas automatizadas [Davis1993]. En [Wieringa 1996], la modificabilidad se considera como una propiedad, junto con la rastreabilidad, de la facilidad de mantenimiento de una especificación. En [IEEE 1996] también se considera la necesidad de mantener las distintas versiones de la especificación que se vayan produciendo debido a los cambios, es decir, que la especificación sea configurable.
8. **Rastreable:** una especificación de requerimientos es rastreable si y sólo si para cada requisito contenido en ella se conoce su origen y puede referenciarse como origen en posteriores documentos durante el desarrollo, es decir, cada requisito puede rastrearse hacia atrás y hacia delante [IEEE 1993, Davis 1993]. Una condición necesaria para que un requisito pueda rastrearse hacia delante es que pueda referenciarse de manera única, normalmente mediante algún tipo de código. La forma habitual de registrar la rastreabilidad son las matrices de rastreabilidad [DoD 1994, Davis 1993]. Como ya se comentó en el punto anterior, en [Wieringa 1996] se considera la rastreabilidad

como una característica necesaria para la mantenibilidad de la especificación, ya que permite conocer el impacto de un cambio al poder saber a qué elementos afecta. En [IEEE 1996] también se considera la rastreabilidad dentro de la propia especificación para poder conocer las relaciones entre los requerimientos, de forma que éstos formen un conjunto enlazado.

9. **Anotada con importancia y estabilidad:** una especificación de requerimientos está anotada con importancia y estabilidad si y sólo si cada requisito contenido en ella está anotado con la importancia que tiene su cumplimiento para clientes y usuarios y la estabilidad que se espera del requisito, es decir, la probabilidad de que cambie durante el desarrollo [Davis 1993, IEEE 1993]. El conocer la importancia de un requisito permite decidir en qué orden implementarlos en el caso de que se hayan acordado entregar versiones sucesivas del producto. Por otro lado, la estabilidad permite a los diseñadores conocer qué grado de flexibilidad deben introducir en el sistema para soportar futuros cambios. Como ya se comentó anteriormente, en [Wieringa 1996] la anotación de los requerimientos con las preferencias de los clientes se considera como una característica de la compleción de la especificación.
10. **Independiente del diseño y la implementación:** una especificación de requerimientos es independiente del diseño y de la implementación si y sólo si no especifica una determinada descomposición del sistema (arquitectura) ni ningún aspecto de su posible implementación [Davis 1993, Wieringa 1996]. Sólo deben admitirse requerimientos que limiten la libertad de los diseñadores y programadores en el caso de que el cliente lo solicite explícitamente para una interpretación de requerimientos innecesarios desde el punto de vista de las restricciones). En esta memoria se utilizará el término rastreabilidad, y no trazabilidad para referirse al término inglés traceability, tal como se propone en [Fernández y Rueda 1998].

---

### **3.2.8 ¿Para qué un Proceso de Ingeniería de Requerimientos?**

El Proceso de ingeniería de requerimientos es un conjunto estructurado de actividades, mediante las cuales obtenemos, validamos y mantenemos el documento de especificación de requerimientos (ESRE).

Las actividades del proceso incluyen la extracción de requerimientos, el análisis, la negociación y la validación.

No existe un proceso único que sea válido de aplicar en todas las organizaciones. Cada organización debe desarrollar su propio proceso de acuerdo al tipo de producto que se esté desarrollando, a la cultura organizacional, y al nivel de experiencia y habilidad de las personas involucradas en la ingeniería de requerimientos. Hay muchas maneras de organizar el proceso de ingeniería de requerimientos y muchas veces tenemos también que recurrir a consultores, ya que ellos tienen una perspectiva mas objetiva que las personas involucradas en el proceso.

Cualquier tarea en donde el resultado sea importante, se puede realizar de mejor manera al utilizar algún tipo de proceso ordenado. Para obtener este orden, diseñamos nuestros procesos basándonos en algún modelo que nos guíe a la hora de diferenciar y secuenciar las actividades.

Por ende, veremos a continuación los modelos aplicables a la Ingeniería de Requerimientos, los analizaremos y veremos cuál se aplica mejor a nuestro escenario, para continuar con el análisis detallado de las diferentes etapas implicadas en este proceso, y las herramientas que mejor se aplican a cada una.

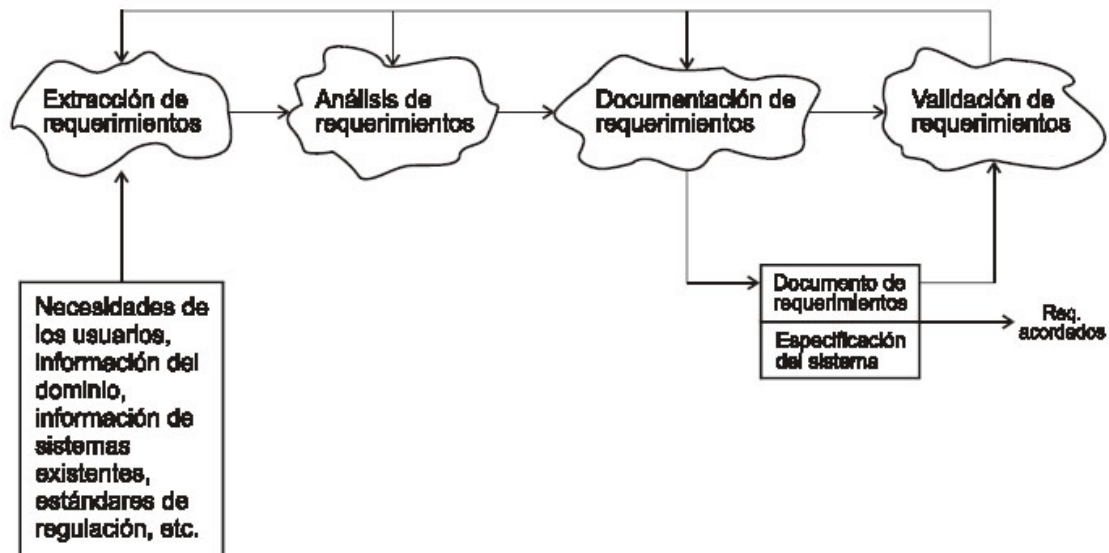
### **3.2.9 Modelo de proceso de IR**

Un modelo es una simplificación de la realidad que incluye aquellos elementos que tienen una gran influencia y omite aquellos elementos que no son relevantes para el nivel de abstracción dado.

En definitiva, los modelos son abstracciones simplificadas y estandarizadas de actividades repetitivas, generalmente producidos desde un punto de vista determinado, por lo que pueden existir diferentes modelos para un mismo proceso.

Sin embargo, en el caso del proceso de IR y desde una perspectiva "intelectual", podemos decir que todos esos diversos modelos parten de una misma base, un modelo "madre" que llamaremos "modelo-abstracto".

Este tipo de modelo nos brinda una vista preliminar del proceso, una secuenciación aproximada y general de las actividades que luego deberemos realizar. Así, presentamos el siguiente ejemplo, en donde cada uno de los compartimientos cubre una sección particular del proceso.



**Figura 3.3: Modelo del Proceso de la Ingeniería de Requerimientos**

Este modelo implica una clasificación secuencial de las actividades, con la elicitación realizada una vez al inicio del proceso. En la realidad, sin embargo, el proceso es iterativo, con estas actividades ejecutadas muchas veces, ya que la tarea de definición de requerimientos no puede definirse por medio de una simple progresión a través de, o relación entre, adquisición, expresión, análisis y especificación. Los requerimientos evolucionan a un paso desigual y tienden a generar requerimientos más extensos a partir de los procesos de definición. Por tanto, la construcción de la especificación de requerimientos es inevitablemente un proceso iterativo. Así, en cada iteración es necesario considerar si la versión actual de la especificación de requerimientos define el requisito del cliente adecuadamente y, si no lo hace, cómo debe cambiarse o debe extenderse más.

Las diversas necesidades de las diferentes organizaciones comienzan a surgir a partir de la aplicación de modelos más detallados. Así, tenemos dos modelos básicos que permiten estudiar el proceso de IR y del cual se derivan numerosas variantes que dependerán del caso de estudio en cuestión.

### 3.2.9.1 Modelo tradicional en cascada

Este modelo sugiere que los resultados de una tarea del proceso llevan a la siguiente, y así sucesivamente. En el ejemplo presentado, la extracción lleva al análisis, el análisis desencadena la documentación, y la documentación inicia la validación.

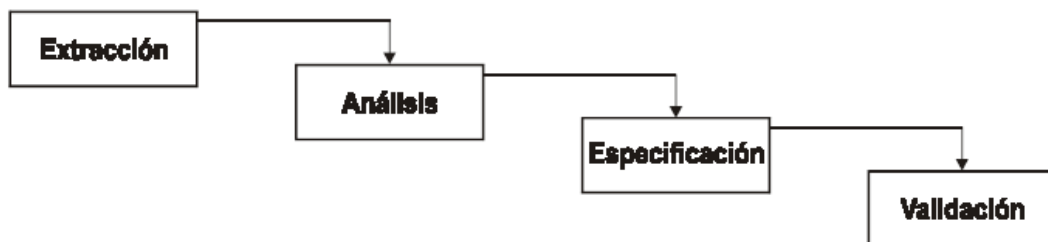


Figura 3.4: Modelo Tradicional en Cascada

Si vemos a este modelo como una descripción general del proceso, es un modelo útil. Sin embargo, debemos entender que la realidad del proceso de IR es mucho más compleja que lo que se vislumbra a partir del modelo en cascada: no existen fases claramente delimitadas ya que hay una retroalimentación constante entre las distintas etapas; los requerimientos del sistema van cambiando por circunstancias ajenas al proceso (como una ley nueva o un cambio de mercado que a su vez cambia las necesidades de la empresa) durante el desarrollo del mismo; se descubren problemas durante la validación que llevan a un cambio de requerimientos, etc.; y todo esto hará que más de una vez tengamos que volver "hacia atrás" en el proceso de IR.

### 3.2.9.2 Modelo en espiral

Un modo alternativo de presentar modelos de actividad que toma en cuenta la retroalimentación entre etapas y la repetición de tareas, es el llamado Modelo en Espiral. [Kotonya G.; Sommerville I. 1998].

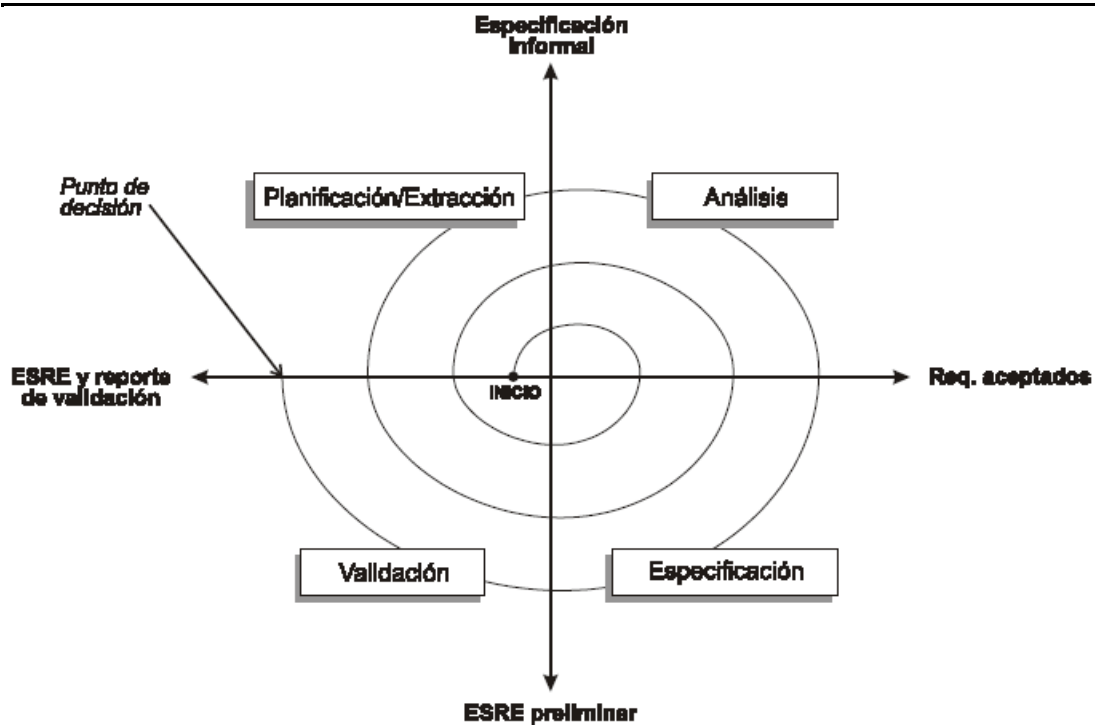


Figura 3.5: Modelo del Ciclo de vida en Espiral

En el dibujo, los cuadros redondeados son tareas. Los cuadrados son productos (inputs u outputs).

En este diagrama, el uso de la espiral implica que las diferentes actividades de la ingeniería de requerimientos son repetidas hasta que se toma la decisión final, que es la aceptación del documento de especificación de requerimientos.

Es decir, si en el diseño preliminar se encuentran problemas, entonces recorreremos el ciclo nuevamente (extracción-análisis-especificación-validación) hasta que todos sean resueltos, que es lo mismo que decir que este ciclo continuará hasta que se pueda elaborar un documento aceptable.

Pero también existen factores externos que pueden determinar la finalización del ciclo, como por ejemplo la presión por cumplir con un determinado cronograma.

Luego del sucinto análisis de los dos modelos básicos antes mencionados, podemos concluir que dado el escenario de trabajo ("el analista se enfrenta a un dominio que desconoce y el cliente presenta un alto grado de incertidumbre con respecto al know-how de todos los procesos de su empresa") es más válida la aplicación del modelo en espiral para desarrollar el proceso de IR. Y es que el modelo en espiral representa de manera más real cómo se irán desarrollando las actividades del proceso; esto es,



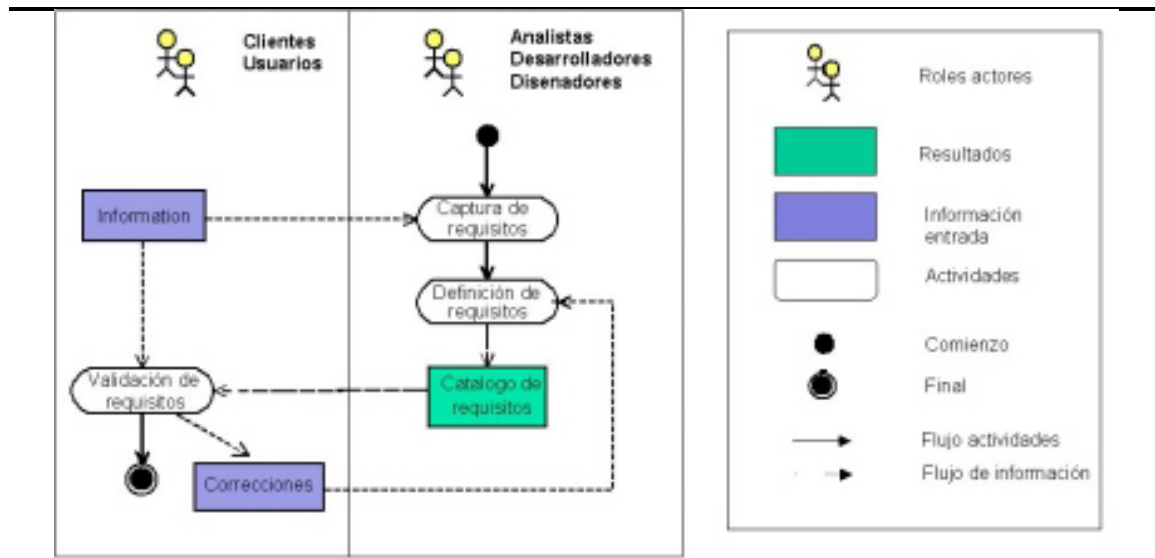
debido al desconocimiento del tema, se genera un grado demasiado alto de incertidumbre que sólo puede disminuirse al repetir el ciclo de trabajo una y otra vez, permitiendo así ajustar todos los parámetros, cada vez en mayor detalle, hasta lograr un resultado satisfactorio.

### **3.2.10 Problemas con el proceso de Ingeniería de Requerimientos**

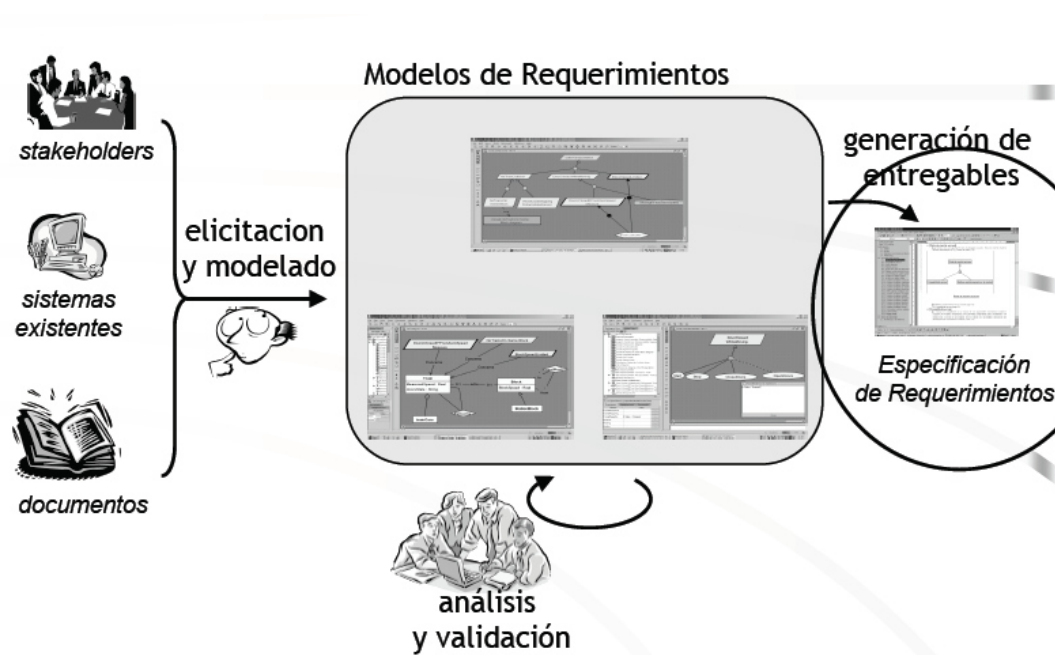
Tendremos problemas si es que:

1. El proceso de requerimientos es **excesivamente largo y costoso**.
2. Los implicados en el proceso se quejan de **falta de tiempo y otros recursos**.
3. Se reciben quejas acerca de la **inteligibilidad del documento** de requerimientos.
4. Los implementadores se quejan del **trabajo perdido por culpa de errores** en los requerimientos.
5. Los usuarios **no utilizan muchas de las capacidades** del sistema.
6. En cuanto el producto se entrega, se recibe una **inmensa cantidad de solicitudes de cambios**.
7. Lleva demasiado **tiempo alcanzar un acuerdo cuando se proponen cambios** a los requerimientos.

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**



**Figura 3.6: Diagrama de actividades del Proceso de la Ingeniería de Requerimientos**



### 3.2.11 Actividades en el Proceso de desarrollo de los requerimientos

Usualmente podemos enumerar las actividades de la ingeniería de Requerimientos en cuatro a saber:

1. Elicitación
2. Análisis y Negociación
3. Especificación
4. Validación

Como toda división de tareas, no es una estricta representación de la realidad, sino que se hace con el fin de sistematizar la realización de la IR. En general la delimitación entre una actividad y la otra no es tan clara, ya que están sumamente interrelacionadas, existiendo un alto grado de iteración y retroalimentación entre una y otra.

#### 3.2.11.1 Elicitación

La elicitación de requerimientos se refiere a la captura y descubrimiento de los requerimientos. Es una actividad más “humana” que técnica. Se identifica a los interesados (*stakeholders*) y se establecen las primeras relaciones entre ellos y el equipo de desarrollo.

#### Fuentes de Requerimientos:

- Metas: Factores críticos de éxito (de la empresa).
- Conocimiento del dominio de la aplicación: Cosas que pueden resultar obvias a los expertos no lo son para los usuarios.
- Los interesados: Los afectados por el sistema.
- El entorno físico que rodea al sistema.
- El entorno organizacional: Los procesos de negocio.

#### Problemas de la Elicitación

- Los interesados a menudo sólo conocen lo que desean en términos muy generales.

- Diferentes interesados tienen requerimientos distintos y los expresan de varias formas.
- Los interesados expresan los requerimientos con sus propios términos y con un conocimiento implícito de su propio trabajo.
- Los usuarios no pueden/saben describir muchas de sus tareas.
- Mucha información importante no llega a verbalizarse.
- La educación no debería ser un proceso pasivo, sino cooperativo.
- El entorno es dinámico, la importancia de los requerimientos puede cambiar, nuevos requerimientos pueden surgir.
- Influencia de factores políticos.

### **3.2.11.2 Análisis y Negociación**

Consiste en **detectar y resolver conflictos entre requerimientos**. Se precisan los límites del sistema y la interacción con su entorno. Se trasladan los requerimientos de usuario a requerimientos del software (implementables).

En el análisis de requerimientos se realizan tres tareas fundamentales:

#### **Modelización Conceptual:**

- Ciertos aspectos de los requerimientos se expresan mediante modelos de datos, de control, de estados, de interacción, de objetos, etc.
- La meta es entender mejor el problema, más que iniciar el diseño de la solución (idealmente).
- El tipo de modelo elegido depende de la naturaleza del problema, la experiencia del modelizador, y de la disponibilidad de herramientas. Por decreto el cliente impone una notación.

#### **Negociación:**

- En todo proceso de IR intervienen distintos individuos con distintos y, a veces, enfrentados intereses.

- Estos conflictos entre requerimientos se descubren durante el análisis. El conflicto no es rechazable, pues los conflictos son fuente de nuevos requerimientos.
- Los conflictos nunca se deben resolver “por decreto”, sino mediante un proceso de (re)negociación.

### **3.2.11.3 Especificación**

En esta fase se documentan los requerimientos acordados con el cliente, en un nivel apropiado de detalle.

En la práctica, esta etapa se va realizando conjuntamente con el análisis, pero podríamos decir que la especificación es el "pasar en limpio" el análisis realizado previamente aplicando técnicas y/o estándares de documentación, como la notación UML.

### **El Documento de Requerimientos**

#### **Tipos de Documentos**

El Documento de Requerimientos es el modo habitual de guardar y comunicar requerimientos. Es buena práctica utilizar, al menos, dos documentos, a distinto nivel de detalle:

- **DRU**: Documento de Requerimientos de Usuario (en inglés, URD).

El DRU se escribe desde el punto de vista del usuario/cliente/interesado. Normalmente los requerimientos de usuario, contenidos en la DRU, no poseen demasiado nivel de detalle. Se incluye la descripción del problema actual (razones por las que el sistema de trabajo actual es insatisfactorio) y las metas que se espera lograr con la construcción del nuevo sistema.

- **ERS**: Especificación de Requerimientos Software (en inglés, SRS).

La ERS desarrolla mucho más los contenidos de la DRU. Los requerimientos del software contenidos en la ERS son, por tanto, más detallados. Contiene la respuesta a la pregunta “¿Qué características debe poseer un sistema que nos permita alcanzar los objetivos, y evitar los problemas, expuestos en la DRU?”.

Un documento es cualquier medio electrónico de almacenamiento y distribución (procesador de textos, base de datos, herramienta de gestión de requerimientos).

### **Características deseables de una ERS**

- No ambigua: Todo requisito posee una sola interpretación.
- Completa: Todo lo que se supone que el software debe hacer está incluido en la ERS.
- Correcta: Todo requisito en la ERS contribuye a satisfacer una necesidad real.
- Comprensible: Todo tipo de lectores entienden la ERS.
- Verificable: Para cada requisito expresado en la ERS existe un procedimiento de prueba finito y no costoso para demostrar que el futuro sistema lo satisface.
- Internamente Consistente: No existen subconjuntos de requerimientos contradictorios.
- Externamente Consistente: Ninguno de los requerimientos está en contradicción con lo expresado en documentos de nivel superior (requerimientos software no pueden contradecir requerimientos de sistema).
- Realizable: Si dados los actuales recursos la ERS es implementable.
- Concisa: La ERS debe ser lo más breve posible sin afectar a los atributos de calidad.
- Independiente del diseño: La ERS se limita a describir el comportamiento externo del sistema.
- Trazable: Cada requisito se puede referenciar de forma unívoca.
- Modificable: Los cambios son fáciles de introducir.
- Electrónicamente Almacenada: En archivo de texto, base de datos, o herramienta de gestión de requerimientos.
- Ejecutable/Interpretable/Prototipable/Animable: Aplicable sólo a ciertas notaciones formales, a partir de las cuales ciertas herramientas software realizan un modelo ejecutable de la ERS.
- Anotada por importancia relativa: Los requerimientos se clasifican según su importancia: Obligatorio, Deseable u Opcional.
- Anotada por estabilidad relativa: Los requerimientos tienen asignada una probabilidad de cambio: Alta, Media o Baja.

- Anotada por versión: Se puede determinar qué requerimientos son satisfechos por qué versión del producto.
- No redundante: Cada requisito se expresa en un solo lugar de la ERS. La redundancia no es mala si aumenta la legibilidad.
- A un nivel de detalle adecuado: Ni demasiado detallada ni demasiado vaga.
- Precisa: La ERS hace uso de valores numéricos para precisar las características del sistema (aplicable a requerimientos no funcionales).
- Reutilizable: Ciertas secciones de la ERS se pueden reutilizar.
- Trazada: Está claro el origen de cada requisito: quién o qué lo pide.
- Organizada: El lector puede encontrar fácilmente la información buscada.
- Con referencias cruzadas: Se utilizan referencias entre requerimientos cruzados (trazabilidad intra-ERS) o entre secciones relacionadas.

Una ERS perfecta es imposible. La calidad de la ERS es muy difícil de cuantificar. En general, una ERS de calidad NO garantiza la ausencia de problemas, pero una ERS pésima garantiza su presencia.

#### **3.2.11.4 Validación de Requerimientos**

El objetivo de la validación de requerimientos es descubrir problemas en el documento de requerimientos antes de comprometer recursos a su implementación. Como resultado de esta validación se produce una *línea-base* (baseline).

El documento debe revisarse para descubrir omisiones, conflictos, ambigüedades, y comprobar la calidad del documento y su grado de adhesión a estándares.

## Validación de Requisitos

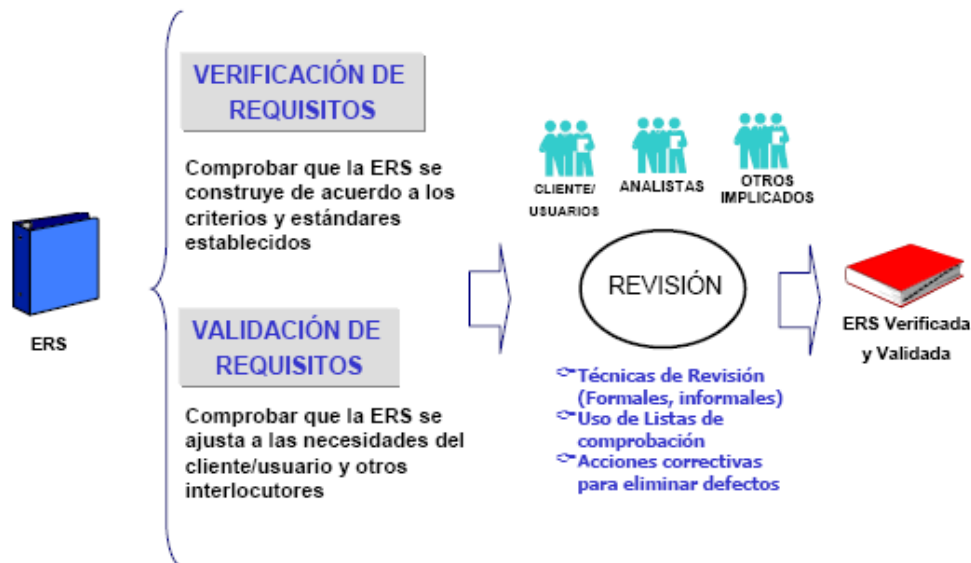


Figura 3.7: Actividad de Validación de Requerimientos

Se deben llevar a cabo diferentes tipos de verificación:

- ✓ Verificación de validez
- ✓ Verificación de consistencia
- ✓ Verificación de integridad
- ✓ Verificación de realismo
- ✓ Verificabilidad

### 3.2.12 Gestión de Requerimientos

Consiste, básicamente, en gestionar los cambios a los requerimientos. Asegura la consistencia entre los requerimientos y el sistema construido (o en construcción). Consume grandes cantidades de tiempo y esfuerzo. Abarca todo el ciclo de vida del producto.

#### 3.2.12.1 Evolución de los requerimientos

Los requerimientos para sistemas software son siempre cambiantes debido a:

- A veces el problema no puede definirse completamente
- Durante el proceso de desarrollo, evoluciona la comprensión del problema
- Una vez que el sistema se ha instalado surgen nuevos requerimientos



- Cuando los usuarios experimentan con el sistema descubren nuevas necesidades y prioridades

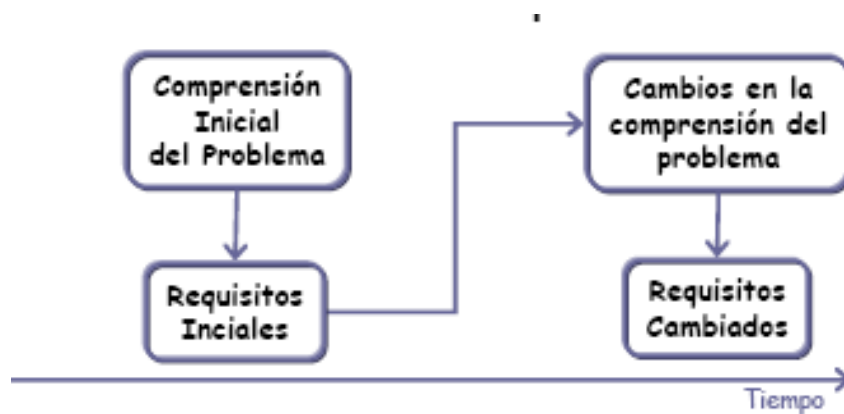


Figura 3.8: Evolución de los requerimientos

Bajo el punto de vista evolutivo, los requerimientos se dividen en dos clases:

1. **Requerimientos duraderos:** se derivan de la actividad principal de la organización y están relacionados con el dominio del problema. (Ej. En un hospital, requerimientos referidos a pacientes, médicos, enfermeras, etc.)
2. **Requerimientos Volátiles:** Requerimientos que probablemente cambien durante el proceso de desarrollo del sistema o después de estar en funcionamiento. (Ej. Requerimientos de políticas gubernamentales del sanidad)

La gestión de Requerimientos es necesaria porque los requerimientos son volátiles, el entorno físico y el entorno organizacional cambian, y la propia existencia del sistema va a generar nuevos requerimientos por parte de los usuarios.

En este sentido, la gestión de requerimientos comprende al conjunto de actividades que intentan entender las necesidades de los usuarios y traducirlas en afirmaciones precisas (no ambiguas), que se usarán en el desarrollo del sistema.

Tomando en cuenta estas definiciones, resulta claro que en la gestión de requerimientos deben participar activamente usuarios, directivos y técnicos, cada uno

con roles y responsabilidades específicas. Si el usuario final no participa del proceso de desarrollo hay más probabilidades de que encuentre que el producto no responde a las necesidades planteadas, lo que podría llevar al fracaso de la implementación. En este sentido, un error bastante habitual es la posición que adoptan los técnicos de no involucrar a los usuarios hasta que el software es visible, es decir, cuando ya fue desarrollado.

#### **3.2.12.2 ¿Qué implica la gestión de Requerimientos?**

1. Definir procedimientos de cambios: definen los pasos y los análisis que se realizarán antes de aceptar los cambios propuestos.
2. Cambiar los atributos de los requerimientos afectados.
3. Mantener la trazabilidad: hacia atrás, hacia delante y entre requerimientos.
4. Control de versiones del documento de requerimientos.

#### **3.2.12.3 Herramientas de Gestión de Requerimientos**

1. Facilitan las tareas relacionadas con la escritura, trazabilidad y gestión de cambios.
2. Organizan los requerimientos en bases de datos.
3. Gestión (incremental) de líneas-base.
4. Todas permiten añadir atributos a los requerimientos.

#### **3.2.12.4 Creación de líneas base**

Una línea base (*baseline*) es un conjunto de requerimientos que, mediante acuerdo entre las partes implicadas, se ha decidido no modificar. Son aquellos requerimientos más importantes que el desarrollador se compromete a implementar.

A lo largo de un proceso de requerimientos pueden crearse (incrementalmente) varias líneas base.

Las herramientas de gestión de requerimientos permiten gestionar fácilmente las líneas base. Por ejemplo, los requerimientos que forman parte de una línea base no podrán ser modificados, salvo por usuarios privilegiados

### **3.2.12.5 Etapas de la gestión de Requerimientos:**

1. Planeación de la administración de requerimientos
2. Administración del cambio de los requerimientos

#### **1. Planeación de la administración de requerimientos**

- La administración de requerimientos es muy cara.
- Durante esta etapa, para cada proyecto, es necesario establecer el nivel de detalle.
- Se tiene que decidir sobre:
  1. La identificación de los requerimientos
  2. Un proceso de administración del cambio
  3. Políticas de rastreo (Trazabilidad)
  4. Ayuda de herramientas CASE

**La trazabilidad** es una propiedad de la especificación de requerimientos que refleja la facilidad de encontrar requerimientos relacionados.

#### **Tipos de información de rastreo (Trazabilidad):**

1. Rastreo de la fuente
  2. Rastreo de los requerimientos
  3. Rastreo del diseño
- La información de rastreo implica utilizar matrices de rastreo.
  - La administración de requerimientos necesita de ayuda automática para:
    1. Almacenar requerimientos
    2. Administrar los cambios
    3. Administrar el rastreo
  - Herramientas CASE, p.e. DOORS y Requisite Pro.

- Sistemas pequeños pueden llevarse con procesador de texto, hoja de cálculo o una pequeña base de datos.

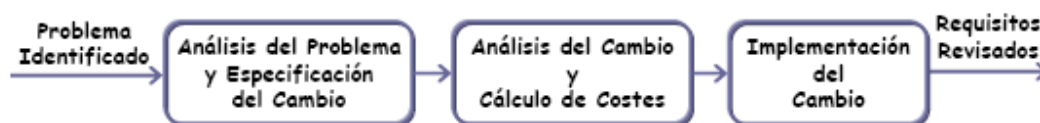
## **2. Administración del cambio de los requerimientos**

Un proceso formal para que todos los cambios propuestos sean tratados de forma consistente.

Etapas:

1. Análisis del problema y especificación del cambio.
2. Análisis cambio y costeo.
3. Implementación del cambio.

Siempre existe la tentación de hacer un cambio urgente al sistema y en retrospectiva modificar el documento de requerimientos. Esto conduce a un desfase e inconsistencias.



**Figura 3.9: Etapas principales del proceso de administración de cambio**

### **Ventajas de la administración del cambio de los requerimientos**

- Todos los cambios propuestos son tratados de forma consistente
- Todos los cambios en el documento de requerimientos se hacen de forma controlada.

### **Definiendo Metodología e Ingeniería de Requerimientos**

La diferencia entre método y metodología, que establece Checkland, es la que se toma como base de la presente propuesta. Dicho autor afirma que: "la esencia de una metodología -en forma opuesta a lo que ocurre en un método o técnica- es que ofrece un conjunto de pautas o principios que en cualquier instancia específica pueden ser ajustadas tanto a las características de la situación en la cual debe ser aplicada como a las personas que usan el enfoque. Es tal la variedad de situaciones problemáticas

humanas que no habrá ningún enfoque para solución de problemas que pueda ser reducido a una fórmula estándar y manejar aún toda la riqueza de las situaciones en particular".

### **3.3 Definición de Términos Básicos**

<b>Termino</b>	<b>Descripción</b>
<b>Abstracción</b>	Consiste en aislar un elemento de su contexto o dentro de los elementos que lo acompañan.
<b>Actividad</b>	Agrupación de tareas que es parte de un proceso
<b>Actor</b>	Es una persona, maquina o sistema que se encarga de interactuar con el sistema para desencadenar un flujo de eventos actividades y acciones dentro del mismo con el objetivo de conseguir un objetivo deseado.
<b>Alcance</b>	Describe todo el trabajo que el proyecto debe incluir y solo el trabajo requerido para completar el proyecto de manera exitosa
<b>Ambiente</b>	Entorno en el que una organización opera. Condiciones externas que afectan un sistema.
<b>Atributo</b>	Información descriptiva asociada a un rasgo específico de un objeto
<b>Bosquejo</b>	Es el primer trazo y no el definitivo de cualquier producción de ingenio. Es una idea vaga de un concepto o plan.
<b>Calidad</b>	Conjunto de propiedades inherentes a un objeto que permiten apreciarlo como igual mejor o peor que el resto de objetos de su misma especie.
<b>Caso de Uso</b>	Forma visual o gráfica de representar la realidad, explica la forma como interactúa el sistema y los actores.
<b>Claridad</b>	Significa que no hay ambigüedad en la especificación

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

	de cada requisito
<b>Ciclo</b>	Una secuencia de instrucciones que se ejecutan en forma repetitiva hasta cumplir con una condición particular.
<b>Compleitud</b>	Se refiere al grado en que un modelo cumple las necesidades de los usuarios en cualquier nivel de abstracción.
<b>Cliente</b>	Toda organización, entidad, comunidad o persona que requiera de los productos o servicios de una organización , consumidor final que utiliza el servicio benefactor.
<b>Consistencia (o coherencia)</b>	Significa que no hay contradicciones entre requerimientos (ni acoplamientos-redundancias)
<b>Fiabilidad</b>	El grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida
<b>Dominio del problema</b>	es el área de conocimiento hacia el cual se orienta cierta solución de software.
<b>DRU</b>	Documento de Requerimientos de Usuario (en inglés, URD).
<b>Entregable</b>	Producto o resultado que debe ser entregada a la persona, empresa u organización interesada
<b>ERS</b>	Especificación de Requerimientos del Software (en inglés, SRS).
<b>Herramienta</b>	Instrumento. Pe un cronograma informal de entrevista o un cuestionario son herramientas para recopilar información. Una herramienta se puede aplicar mediante distintos métodos o técnicas.
<b>IR</b>	Ingeniería de Requerimientos
<b>RF</b>	Abreviatura de requerimiento funcional.
<b>RNF</b>	Abreviatura de requerimiento no funcional
<b>Trazabilidad</b>	Es la correspondencia entre cada requisito del

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

	software y /o uno o más requerimientos del usuario, u otras fuentes (trazabilidad hacia atrás) o una o varias partes del diseño o la implementación
<b>Técnica</b>	La técnica es el procedimiento o el conjunto de procedimientos que tienen como objetivo obtener un resultado determinado, ya sea en el campo de la ciencia, de la tecnología o en otra actividad.
<b>UML</b>	Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por la OMG.
<b>StakeHolder</b>	Literalmente, stakeholder significa el que tiene una estaca, aunque el significado real se deduce de la expresión to have a stake in something (tener una estaca en algo) que significa tener intereses en algo. Por lo tanto, un stakeholder es cualquier persona que tenga intereses en juego durante la fase de ingeniería de requerimientos: clientes, usuarios, ingenieros de requerimientos, desarrolladores, etc.

#### **4. METODOLOGIA DE LA INVESTIGACION**

En este trabajo se describe un entorno metodológico para la ingeniería de Requerimientos que esta conformado por:

1. Un modelo de procesos iterativo en el que se identifican cuatro actividades principales: elicitación, análisis, documentación y validación.
2. Una metodología para la elicitación de requerimientos de sistemas de información, incluyendo las tareas a realizar, los productos a obtener, las técnicas y herramientas a emplear, principalmente plantillas de requerimientos.
3. Una metodología para el análisis de requerimientos de sistemas de información, incluyendo las tareas a realizar, los productos a obtener, las técnicas y herramientas a emplear, basadas en el estándar UML y con relaciones de rastreabilidad hacia los productos de la actividad anterior que facilita la reutilización de elementos complejos.
4. Una metodología para la documentación de requerimientos incluyendo tareas a realizar, los productos a obtener, las técnicas y herramientas a emplear.
5. La estructura del Documento de Requerimientos que será el producto entregable para el cliente el cual se entrega en fechas previamente acordadas. Este documento define de forma precisa el producto de software que se va a construir. Las decisiones hechas escribiendo el documento están basados en información que proviene de la actividad de elicitación y análisis, es decir requerimientos analizados con los usuarios para satisfacer el diseño del sistema.



6. Las definiciones de las técnicas recomendadas en cada actividad del Proceso de Ingeniería de requerimientos
7. Las Plantillas propuestas para la captura y análisis de requerimientos
8. Presentamos un caso práctico de un sistema logístico de una distribuidora de alimentos en su modulo de almacén, utilizando algunas tareas recomendadas con sus plantillas y diagramas a modo de aplicación de la metodología propuesta.

La estructura de este documento es la siguiente: en la sección 4.1 se describen la metodología de la actividad de elicitación, en la sección 4.2 se describen la metodología de la actividad de análisis de requerimientos, en la sección 4.3 se define la metodología de la actividad de documentación de requerimientos, en la sección 4.5 se define la estructura del documento de requerimientos, en la sección 4.6 se describen las técnicas recomendadas para obtener el producto, en la sección 4.7 se definen las plantillas utilizadas en la metodología y por último en la sección 4.8 se incluye un caso práctico de la aplicación de esta metodología en un sistema logístico en su modulo de almacén.

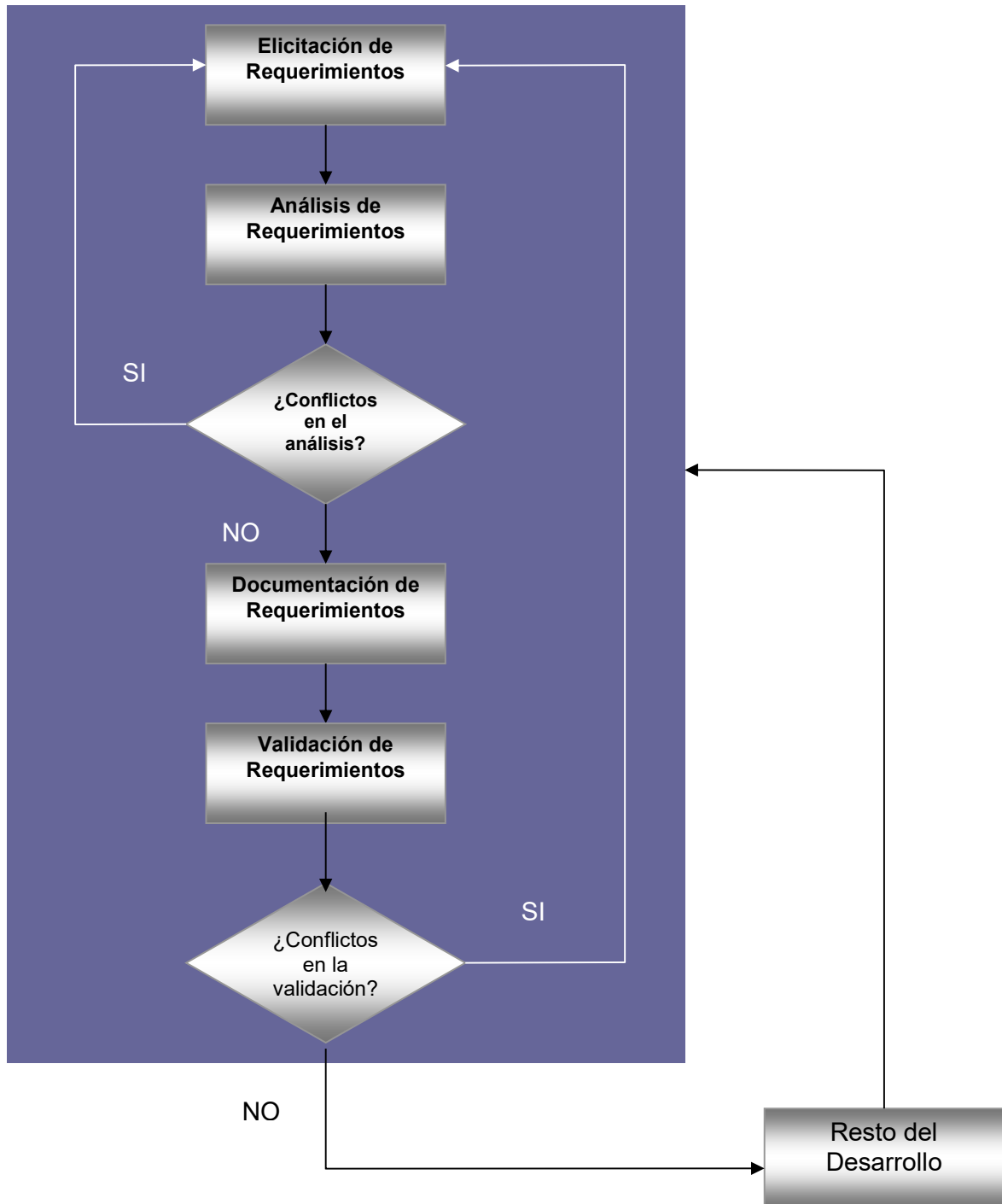
En la tabla 4.1 se resumen esquemáticamente los problemas que se abordan en este trabajo y en qué actividad del modelo de procesos propuesto serán tratados.

**Tabla 4.1 : Problemas Abordados vs. Actividades del Modelo de Proceso de la IR**

<b>Problemas Ingeniería de Requerimientos</b>		<b>Actividad</b>	<b>Abordado?</b>
1	<b>Comunicación con Clientes y usuarios</b>	Elicitación	✓
	<b>Eliminación de ambigüedades</b>	Análisis	✓
	<b>Priorización de requerimientos</b>	Elicitación	✓
	<b>Determinar el ámbito del Sistema</b>	Elicitación	✓
	<b>Estimación y planificación</b>	No contemplado	X
	<b>Asegurar la Compleción</b>	No contemplado	X
2	<b>Integración de múltiples vistas</b>	Análisis	✓
	<b>Evaluación de Alternativas</b>	No contemplado	X
	<b>Obtener especificaciones de calidad</b>	Análisis	✓
	<b>Validación</b>	No contemplado	X
	<b>Proceso sin costuras</b>	Análisis	✓
3	<b>Mantenibilidad de Requerimientos</b>	No contemplado	X
	<b>Reutilización de requerimientos</b>	Elicitación	✓
	<b>Ingeniería inversa de requerimientos</b>	No contemplado	X

#### 4.1 Propuesta de modelo de procesos de ingeniería de requisitos

Figura 4.1: Modelo Propuesto de Proceso de la Ingeniería de Requerimientos  
Iteración de actividades



Es fundamental establecer un modelo de procesos para llegar a comprender los diversos problemas de la ingeniería de requerimientos así como para poder desarrollar metodologías y herramientas de desarrollo. El modelo de procesos de ingeniería de requerimientos que se propone en este trabajo consta de cuatro actividades que son principales, elicitación, análisis, documentación y validación, y su principal característica es la iteratividad, tal como puede verse en la figura 4.1

Al igual que la mayor parte de las normas y autores consultados, asumimos que el proceso de elicitar requerimientos, analizarlos y validarlos es iterativo por naturaleza, ya que es prácticamente imposible obtener todos los requerimientos y que éstos tengan las propiedades deseables de un requerimiento sin tener que volver atrás en algún momento del proceso. Sin embargo, no por ello se debe dejar de intentar elicitar, y posteriormente analizar, documentar y validar, la mayor cantidad posible de requerimientos en cada iteración, intentando alcanzar la situación ideal en la que el proceso sería lineal.

#### **4.1.1 Ciclos de iteración**

En el modelo propuesto se definen cuatro posibles ciclos de iteración, dos Internos al proceso y uno externo.

##### **4.1.1.1. Ciclo elicitación–análisis–documentación–validación**

El ciclo principal **elicitación–análisis–documentación–validación** es un ciclo interno que indica la posibilidad de que durante el proceso de validación de los requerimientos por parte de los clientes y usuarios aparezcan conflictos o nuevos requerimientos que hasta entonces estaban ocultos. En esas circunstancias, es necesario resolver dichos conflictos y consensuar los nuevos requerimientos mediante nuevas reuniones de elicitación/negociación, repitiendo a continuación las actividades de análisis, documentación y validación.

#### **4.1.1.2. Ciclo elicitación–análisis**

El segundo ciclo interno, elicitación–análisis, se trata de un ciclo que indica la posibilidad de que durante la realización del análisis de los requerimientos elicitados se descubran conflictos o deficiencias en dichos requerimientos, lo que puede provocar la necesidad de nuevas reuniones de elicitación/ negociación y el posterior análisis de sus resultados.

#### **4.1.1.3. Ciclo ingeniería de requerimientos–resto del desarrollo**

El tercer ciclo, entre la ingeniería de requerimientos y el resto del desarrollo, muestra la posibilidad de que durante el resto del desarrollo sea necesario volver a alguna de las actividades de ingeniería de requerimientos, posiblemente porque se detecte la necesidad de renegociar algunos requerimientos de difícil implementación, porque aparezcan nuevos requerimientos durante el desarrollo, etc.

Después de definidas las actividades del modelo propuesto del proceso de la Ingeniería de requerimientos, Procedemos a enumerar las actividades en las que se basa la metodología propuesta:

- 1 Elicitación de requerimientos**
- 2 Análisis de Requerimientos**
- 3 Documentación de Requerimientos**
- 4 Validación de Requerimientos**

#### **4.2 Propuesta de la metodología para la elicitación de requerimientos**

Las tareas recomendadas para obtener el Documento de Requerimientos en esta metodología son las siguientes:

1. **Tarea1:** Formar el equipo multidisciplinario.
2. **Tarea2:** Obtener información sobre el dominio del problema y el sistema actual.
3. **Tarea3:** Preparar y realizar las reuniones de elicitación/negociación.
4. **Tarea4:** Identificar/revisar los objetivos del sistema.
5. **Tarea5:** Clasificar requerimientos.
6. **Tarea6:** Evaluar y racionalizar.
7. **Tarea7:** Dar prioridad.
8. **Tarea8:** Integrar y validar.

#### **4.2.1 Tarea 1: Formar el equipo multidisciplinario**

##### **Objetivos**

- ✓ Identificar personal capacitado para escoger y desarrollar las técnicas apropiadas de acuerdo al factor humano de la situación actual.

##### **Descripción**

Considerando que la formación de la gente de sistemas, tratándose de problemas con alta incidencia del factor humano, no tiene la especialización necesaria como para diagnosticar el método de elicitación más apropiado para cada caso en particular, se aconseja que la recolección de requerimientos sea efectuada con el asesoramiento de profesionales especializados. Este asesoramiento puede extenderse incluso a un liderazgo activo de las sesiones de elicitación por parte de especialistas en ciencias de la comunicación o en ciencias del conocimiento.

En esta área el principal objetivo es sumar adecuadamente personas a cargos vacantes en un equipo de desarrollo de sistemas Software, o bien la conformación de un equipo. En este momento hay que establecer cuáles son las condiciones deseables para dicho equipo. Y para poder establecer estas condiciones deseables se debe tener un adecuado conocimiento de como actúan los equipos de trabajo. Este conocimiento, necesario para poder realizar la selección de las personas adecuadas, es responsabilidad de los administradores de Software y futuros jefes de proyecto, de acuerdo con las políticas de la empresa.

De esta forma, con relación a los equipos de trabajo se debe tener en cuenta:

- Los equipos de trabajo están conformados por individuos, y estos individuos interactúan entre sí. El administrador debe entender las limitaciones humanas e intentar no llevar a cabo proyectos irrealizables.
- Los sistemas de Software son utilizados por personas. Si las limitaciones y capacidades de los desarrolladores no son tomadas en cuenta, difícilmente los sistemas serán bien aprovechados por los usuarios.
- De esta forma la selección es importante teniendo en cuenta muchos factores de la personalidad de los postulantes. Sin embargo, no sólo se debe tener en cuenta esta vista al momento de escoger, sino también considerar

experiencias anteriores, habilidades en ciertos aspectos del desarrollo, etc. Para tener éxito en la selección de los futuros integrantes de los equipos de trabajo, basarse sólo en la personalidad de los postulantes puede ser una mala decisión, debido a que:

- La personalidad es dinámica, no estática. Esta puede cambiar en el transcurso de una carrera profesional o por cambios en el ambiente de trabajo.
- Diferentes personalidades pueden resultar adecuadas a diferentes aspectos del desarrollo, tales como diseño, testing y otras.
- Los postulantes pueden hacer trampa al completar los tests de personalidad, llenándolos no con sus datos reales, sino con lo que piensan que podría ser la personalidad más adecuada al trabajo. Por esto la realización de una buena entrevista es vital para la selección.

### **Productos entregables**

- ✓ Participantes en el proyecto como parte del Documento de Requerimientos.

### **Técnicas recomendadas**

- ✓ Entrevistar a las personas candidatas para el desarrollo de la elicitación evaluándolas según la experiencia con el factor humano.

### **Herramientas recomendadas**

- ✓ Test de Personalidad, Test de Conocimientos.



#### **4.2.2 Tarea 2: Obtener información sobre el dominio del problema y el Sistema Actual**

##### **Objetivos**

- ✓ Conocer el dominio del problema.
- ✓ Conocer la situación actual.

##### **Descripción**

Es de amplia difusión que para definir los requerimientos de un sistema de software, se requiere un profundo conocimiento del Dominio. Esto incluye no sólo la visión actual del mismo, sino también los planes para el futuro. A menos que el ingeniero de requerimientos tenga un conocimiento previo del dominio, puede suceder que se vea sobrecargado de datos de diferente naturaleza.

Para afrontar este problema, es aconsejable utilizar un proceso gradual que evolucione desde el aprendizaje a la definición. Durante la fase de aprendizaje, el foco debe ubicarse en los hechos actuales que deberían ser organizados y registrados. Luego, deben comprenderse los planes a futuro; pero en lugar de registrarlos tal como se los obtiene, deberían ser usados para diseñar modelos del dominio futuro.

En el ámbito de Ingeniería Es fundamental conocer el dominio del problema y los contextos organizacional y operacional, además de identificar quiénes son los involucrados en la toma de decisiones, tanto como quién se verá afectado por la formulación de los problemas y la eventual solución, es decir, conocer la situación actual. Enfrentarse a un desarrollo sin conocer las características principales ni el vocabulario propio de su dominio suele provocar que el producto final no sea el esperado por clientes ni usuarios. Por otro lado, mantener reuniones con clientes y usuarios sin conocer las características de su actividad hará que probablemente no se entiendan sus necesidades y que su confianza inicial hacia el desarrollo se vea deteriorada enormemente.

Esta tarea es opcional, ya que puede que no sea necesario realizarla si el equipo de desarrollo tiene experiencia en el dominio del problema y el sistema actual es conocido.

### **Productos entregables**

- ✓ Introducción
- ✓ participantes en el proyecto, principalmente clientes y desarrolladores.
- ✓ descripción del sistema actual.
- ✓ glosario de términos
- ✓ declaración del contexto del problema, de los objetivos globales.
- ✓ límites e interfaces para el sistema original, como parte del Documento de Requerimientos.

### **Técnicas recomendadas**

En el caso de que se trate de un dominio muy específico puede ser necesario recurrir a:

### **Técnicas recomendadas**

- ✓ Estudio de documentación.
- ✓ DCO
- ✓ Entrevista
- ✓ Modelado del sistema actual.
- ✓ Sistemas Existentes

### **Herramientas recomendadas**

- ✓ Arqueología de documentos
- ✓ Construcción de glosarios de términos
- ✓ Cadena de valor
- ✓ Análisis FODA
- ✓ Cuestionarios.

#### **4.2.3 Tarea 3: Preparar y realizar las reuniones de elicitación/negociación.**

##### **Objetivos**

- ✓ Identificar a los usuarios participantes.
- ✓ Conocer las necesidades de clientes y usuarios.
- ✓ Resolver posibles conflictos.

##### **Descripción**

Teniendo en cuenta la información recopilada en la tarea anterior, en esta tarea se deben preparar y realizar las reuniones con los clientes y usuarios participantes con objeto de obtener sus necesidades y resolver posibles conflictos que se hayan detectado en iteraciones previas del proceso. Esta tarea es especialmente crítica y ha de realizarse con especial cuidado, ya que generalmente el equipo de desarrollo no conoce los detalles específicos de la organización para la que se va a desarrollar el sistema y, por otra parte, los clientes y posibles usuarios no saben qué necesita saber el equipo de desarrollo para llevar a cabo su labor.

##### **Productos entregables**

- ✓ Participantes en las sesiones de elicitación.
- ✓ Requerimientos o conflictos, que se hayan identificado claramente durante las sesiones de elicitación.

##### **Técnicas recomendadas**

- ✓ Entrevistas
- ✓ Joint Application Development JAD
- ✓ Observación(Etnografía)
- ✓ Sistemas Existentes
- ✓ Aprendiz

##### **Herramientas recomendadas**

- ✓ Casos de Uso
- ✓ Cuestionarios
- ✓ Run Use Case Workshop (talleres)

- ✓ Grabaciones de Video y Audio
- ✓ Brainstorming
- ✓ Arqueología de Documentos
- ✓ Prototipos
- ✓ Glosario

Con estas técnicas y herramientas se incluye la utilización de plantillas de objetivos, plantillas de requerimientos funcionales, plantillas requerimientos no funcionales, y plantillas de conflictos descritas más adelante que pueden usarse directamente durante las sesiones de elicitación.

#### **4.2.4 Tarea 4: Identificar/revisar los objetivos del sistema.**

##### **Objetivos**

- ✓ Identificar los objetivos que se deseen conseguir mediante el sistema software a desarrollar.
- ✓ Revisar, en el caso de que haya conflictos, los objetivos previamente identificados.

##### **Descripción**

En esta tarea se deben de obtener qué objetivos se esperan alcanzar en el sistema a desarrollar cuando este ya este en uso y revisarlos en función de los conflictos identificados.

Puede que los objetivos hayan sido proporcionados antes de comenzar el desarrollo.

##### **Productos entregables**

- ✓ Objetivos del sistema como parte del Documento de Requerimiento del Sistema.

##### **Técnicas recomendadas**

- ✓ Análisis de factores críticos de éxito [MAP 1995].
- ✓ Plantilla para especificar los objetivos del sistema.

#### 4.2.5 Tarea 5: Clasificar / Revisar Requerimientos

##### Objetivos

- ✓ Identificar los requerimientos de almacenamiento de información
- ✓ Identificar requerimientos funcionales
- ✓ Identificar requerimientos no funcionales
- ✓ Identificar requerimientos de restricciones o reglas de negocio que deberá cumplir el sistema software a desarrollar.
- ✓ Identificar los actores del sistema del software a desarrollar en el caso de los requerimientos funcionales
- ✓ Expresar los requerimientos funcionales en casos de uso

##### Descripción

En esta tarea se debe identificar la información relevante para el cliente la cual gestionara y almacenara: **requerimientos de almacenamiento**, qué debe hacer el sistema con esta información: **requerimientos funcionales**, qué requerimientos son de carácter técnico o legal: **requerimientos no funcionales**; así como qué restricciones o reglas de negocio debe cumplir dicha información.

En el caso de los requerimientos funcionales además se debe identificar a las personas o sistemas que serán los orígenes o destinos de la información que consumirá o producirá y que forman su entorno del sistema a desarrollar: **actores**; e identificar los casos de uso asociados a los actores, los pasos de cada caso de uso y posteriormente se detallar los casos de uso con las posibles excepciones hasta definir todas las situaciones posibles.

En el caso de que se considere necesario, se podrá optar por expresar algunos o todos los requerimientos funcionales de la forma tradicional, es decir, mediante un párrafo en lenguaje natural, en lugar de hacerlo mediante casos de uso, y realizar esta tarea en la actividad de documentación.

En el caso de los requerimientos no funcionales se deben clasificar según: Requerimientos de comunicaciones del sistema, Requerimientos de interfaz de

usuario, Requerimientos de fiabilidad, Requerimientos de entorno de desarrollo, Requerimientos de portabilidad, etc., según la situación actual.

### **Productos entregables**

- ✓ Requerimientos de almacenamiento de información.
- ✓ Requerimientos funcionales.
- ✓ Requerimientos no funcionales
- ✓ Requerimientos de restricciones

### **Técnicas recomendadas**

- ✓ Plantillas

### **Herramientas**

- ✓ Plantilla para requerimientos de almacenamiento de información
- ✓ Plantilla para requerimientos funcionales
- ✓ Plantilla para requerimientos no funcionales
- ✓ Plantilla para requerimientos de restricciones de información
- ✓ Casos de uso

## **4.2.6 Tarea 6: Evaluar y racionalizar**

### **Objetivos**

- ✓ Evaluar la coherencia de la información
- ✓ Evaluar la inconsistencia de la información.
- ✓ Identificar conflictos entre los requerimientos previamente identificados.

### **Descripción**

A partir de la información obtenida en las tareas anteriores, y teniendo en cuenta los objetivos identificados, se evalúan todos los requerimientos.

Se debe tener una valoración del riesgo, para encaminar las inquietudes técnicas, de costos y de tiempo es decir debe examinarse que la información reunida en las tareas anteriores sea coherente para determinar si los requerimientos verdaderos están escondidos o expresados explícitamente.

Mediante la evaluación de la información de requerimientos se ponen en evidencia las inconsistencias que pueden surgir entre los requerimientos y a la vez los conflictos que se dan entre los requerimientos.

#### **Productos entregables**

- ✓ Requerimientos inconsistentes, incoherentes o poco entendibles
- ✓ Conflictos detectados

#### **Técnicas recomendadas**

- ✓ Se realizan preguntas del tipo ¿Por qué usted necesita X?, y si esta pregunta tiene una respuesta concreta, entonces es un requerimiento, si no es un falso requerimiento.

### **4.2.7 Tarea 7: Dar Prioridad**

#### **Objetivos**

- ✓ Definir las necesidades más importantes y abordarlas primero.

#### **Descripción**

Teniendo ya los requerimientos consistentes, se da un orden de prioridades, de manera tal que las necesidades de alta prioridad pueden ser tomados primero, lo que permite definirlos y reexaminar los posibles cambios de los requerimientos, antes que los requerimientos de baja prioridad (que también pueden cambiar) sean implementados. Durante el desarrollo del sistema, esto permite una disminución de los costos y ahorro de tiempo en procesamiento de los inevitables cambios de los requerimientos. Los requerimientos deben tener prioridades basándose en las necesidades del usuario, el costo y la dependencia.

#### **Productos entregables**

- ✓ Lista de Requerimientos con prioridades.

#### **Técnicas recomendadas**

- ✓ Sistemas Existentes
- ✓ Entrevistas
- ✓ Objetivos

#### **4.2.8 Tarea 8: Integrar y Validar**

##### **Objetivos**

- ✓ Obtener un conjunto de requerimientos, expresados en un lenguaje entendible para el usuario.
- ✓ Validar la consistencia con respecto a las metas organizacionales obtenidas en la primera etapa.
- ✓ Validar los requerimientos según todas las partes afectadas

##### **Descripción**

Esta tarea se lleva a cabo de manera tal que sea posible obtener un conjunto de requerimientos, entendibles por el usuario, de los cuales se pueda validar la consistencia con respecto a los objetivos organizacionales. Las tareas de integración deben ser ejecutadas principalmente por el analista de sistemas, y los resultados del proceso de elicitación comunicarlos a las otras comunidades involucradas. Esta validación de los requerimientos realizada por todas las partes afectadas, asegura que se alcanza lo deseado.

##### **Productos entregables**

- ✓ Lista de Requerimientos validados por todos los usuarios relacionados con el Sistema a desarrollar.

##### **Técnicas recomendadas**

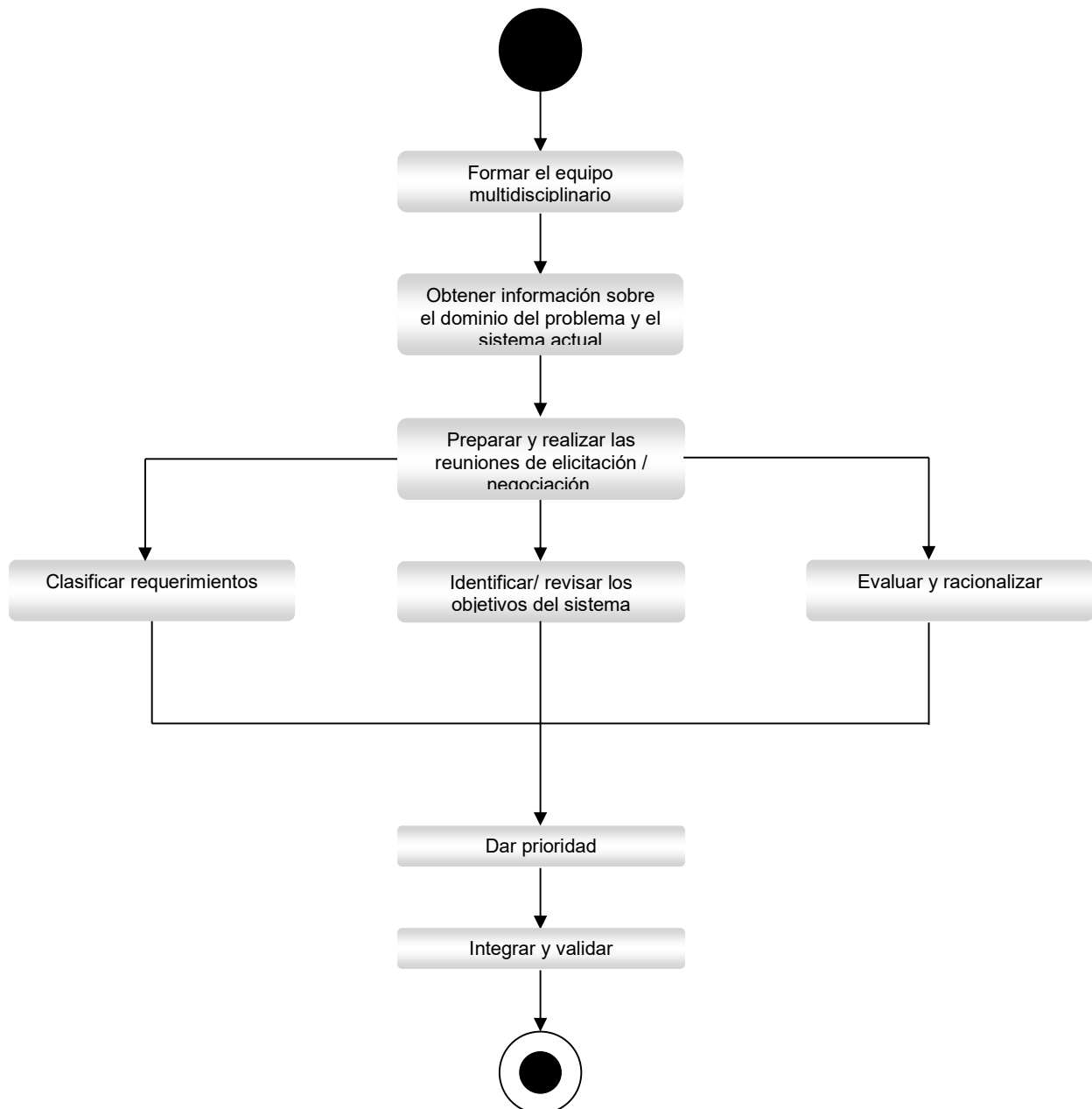
- ✓ Entrevistas

Como es de esperar, a los efectos de obtener buenos requerimientos, todos estos pasos deben iterar ante la menor inconsistencia detectada, aconsejándose que la iteración se realice recurriendo al cliente/usuario, tantas veces como sea necesario, para garantizar una correcta depuración del producto final de la etapa de elicitación.



Si se hiciera una representación gráfica de la propuesta metodológica para la etapa de elicitación de requerimientos, la misma quedaría bosquejada de la manera que se puede apreciar en la figura 4.2

**Figura 4.2: Diagrama de Actividades de la actividad de Elicitación del Proceso de la Ingeniería de Requerimientos**



#### **4.3 Propuesta de la Metodología para el Análisis de Requerimientos**

Usualmente se hace un análisis luego de haber producido un bosquejo inicial del documento de requerimientos; aquí se leen los requerimientos, se conceptúan, se investigan, se intercambian ideas con el resto del equipo, se resaltan los problemas, se buscan alternativas y soluciones, y luego se van fijando reuniones con el cliente para discutir los requerimientos.

Los pasos a realizar durante esta etapa tienen como objetivo la obtención de buenos requerimientos, para asegurar de esta manera que lo que se derivará a las etapas posteriores será de calidad tal que permita que disminuyan las fallas de los sistemas.

Debemos destacar lo difícil que es convertir el análisis en un proceso estructurado y sistemático, ya que es una etapa bastante "subjetiva" porque depende en gran medida del juicio y de la experiencia del analista.

Las Tareas recomendadas durante esta actividad son:

1. **Tarea 1:** Reducir ambigüedades en los requerimientos.
2. **Tarea 2:** Traducir a lenguaje técnico los requerimientos.
3. **Tarea 3:** Analizar los requerimientos de almacenamiento de información.
4. **Tarea 4:** Analizar los requerimientos funcionales.
5. **Tarea 5:** Analizar los requerimientos no funcionales.
6. **Tarea 6:** Desarrollar prototipos.

#### **4.3.1 Tarea 1: Reducir ambigüedades en los requerimientos.**

##### **Objetivos**

- ✓ Reducir la ambigüedad del lenguaje del usuario.

##### **Descripción**

Los requerimientos obtenidos como resultado final de la etapa de elicitación, deben ser tratados a efectos de llevarlos a una notación que permita reducir la ambigüedad del lenguaje del usuario. Por consiguiente, en esta tarea se debe eliminar los términos que tienen más de una acepción.

En esta etapa se realizará una reunión interna de los integrantes del grupo, para analizar los requerimientos obtenidos, se realizan intercambios de ideas acerca de los términos con mas de una acepción para luego remplazarlos por términos adecuados que serán parte del glosario de términos del dominio,

##### **Productos entregables**

- ✓ Conflictos detectados.
- ✓ Glosario de Términos

##### **Técnicas**

- ✓ Reunión del Grupo. Preguntas generales, abiertas.

##### **Herramientas**

- ✓ Glosario de Términos
- ✓ Plantilla de Requerimientos de Conflictos.

#### **4.3.2 Tarea 2: Traducir a lenguaje técnico los requerimientos.**

##### **Objetivos**

- ✓ Aproximar los términos del usuario a los términos del sistema de software.

##### **Descripción**

Los requerimientos, ya con menos ambigüedades, deben ser llevados a un lenguaje que se vaya aproximando al lenguaje técnico. Mediante esta traducción se busca aproximar los términos del usuario a los términos del sistema de software.

##### **Productos entregables**

- ✓ Listado Requerimientos (traducidos)
- ✓ Glosario de Términos Técnicos.

##### **Técnicas**

- ✓ Reunión de Grupo, Revisión y traducción de los requerimientos del sistema.

##### **Herramientas**

- ✓ Glosario de Términos del Dominio.

### **4.3.3 Tarea 3: Analizar los requerimientos de almacenamiento de información**

#### **Objetivos**

- ✓ Detectar posibles conflictos en los requerimientos de información, tanto en los de almacenamiento como en los de restricciones.
- ✓ Profundizar en el conocimiento de los requerimientos de almacenamiento de Información.

#### **Descripción**

El objetivo principal de esta tarea, al igual que el de la siguiente, es descubrir conflictos en los requerimientos de almacenamiento profundizando en el conocimiento del problema y estableciendo las bases para un futuro diseño del sistema.

La forma habitual de alcanzar estos objetivos es mediante la construcción de modelos abstractos. En esta propuesta se recomienda la utilización de técnicas de modelado orientadas a objetos.

Los productos resultantes de la realización de esta tarea son el modelo estático del sistema, compuesto por los diagramas de clases y la descripción textual de los elementos que lo integran, y los posibles conflictos que se detecten al construir el modelo.

#### **Productos entregables**

- ✓ Diagramas de clases como parte del Documento de Requerimientos.
- ✓ Conflictos detectados

### **Técnicas**

- ✓ Diagrama de clases

### **Herramientas**

- ✓ Plantillas para las clases.
- ✓ Plantillas para asociaciones entre clases.
- ✓ UML - RATIONAL ROSE

## **4.3.4 Tarea 4: Analizar los requerimientos funcionales del Sistema**

### **Objetivos**

- ✓ Detectar posibles conflictos en los requerimientos funcionales, tanto en los expresados de forma tradicional como en los expresados mediante casos de uso.
- ✓ Profundizar en el conocimiento de los requerimientos Funcionales.

### **Descripción**

Esta tarea es similar a la anterior, con la diferencia de que se centra en los requerimientos funcionales, expresados mediante casos de uso, en lugar de hacerlo en los requerimientos de almacenamiento de información.

Para analizar los requerimientos funcionales lo habitual es construir modelos funcionales y, si se considera oportuno, modelos dinámicos. Por lo tanto, los productos resultantes de la realización de esta tarea son los modelos funcionales y dinámicos y los conflictos que se hayan detectado al construir ambos modelos.

Se propone utilizar Diagramas del UML que son los Diagramas de comportamiento Y Diagrama de Interacción para modelar los aspectos funcionales del sistema.

### **Productos entregables**

- ✓ Diagramas de Comportamiento y de Interacción según UML
- ✓ Conflictos detectados durante el análisis

### **Técnicas**

- ✓ Diagrama de actividades
- ✓ Diagrama de casos de uso
- ✓ Diagrama de estados
- ✓ Diagrama de secuencia

### **Herramienta**

- ✓ UML - RATIONAL ROSE

## **4.3.5 Tarea 5: Analizar los requerimientos no funcionales del Sistema**

### **Objetivos**

- ✓ Detectar posibles conflictos en los requerimientos no funcionales,
- ✓ Profundizar en el conocimiento de los requerimientos.

### **Descripción**

Esta tarea tiene también como objetivo descubrir conflictos en los requerimientos, en este caso en los requerimientos no funcionales. Sin embargo, a diferencia de las dos tareas anteriores, las escasas técnicas que integran este tipo de requerimientos aun no pueden considerarse satisfactorias

La naturaleza heterogénea de este tipo de requerimientos hace que su análisis sea necesario realizarlo mediante una lectura detenida de su contenido, y combinando esta lectura con la experiencia, detectar posibles conflictos como la imposibilidad técnica de la implementación de ciertos requerimientos, la necesidad de optar por unas características u otras (por ejemplo flexibilidad frente a eficiencia), etc.

Normalmente, estos requerimientos se tendrán en consideración durante el diseño de la arquitectura del sistema, lo que probablemente provocará iteraciones entre el resto del desarrollo y la fase de ingeniería de requerimientos.

Los productos resultantes de esta tarea son, por lo tanto, aquellos conflictos que se hayan detectado al realizar el análisis de los requerimientos no funcionales.

#### **Productos entregables**

- ✓ Conflictos detectados durante el análisis.

#### **Técnicas**

- ✓ Conflictos detectados durante el análisis.

#### **Herramientas**

- ✓ Plantilla de Requerimientos de Conflictos
- ✓ Sistemas Existentes.
- ✓ Prototipos.

### **4.3.6 Tarea 6: Desarrollar prototipos**

#### **Objetivos**

- ✓ Obtener requerimientos relativos a la interfaz de usuario
- ✓ Detectar posibles conflictos en los requerimientos
- ✓ Profundizar en el conocimiento de los requerimientos

#### **Descripción**

En esta tarea se deben desarrollar prototipos que permitan tanto a los clientes y usuarios como a los desarrolladores tener una idea más clara del sistema a desarrollar e identificar nuevos requerimientos o conflictos que hayan permanecidos ocultos hasta el momento. Lo más habitual es que el prototipo sea desechable, es decir, que una vez que se haya utilizado no se desarrolle tomando su código como base.



El objetivo de esta tarea es el desarrollo de prototipos que puedan utilizarse durante la elicitación o validación de los requerimientos, y que por lo tanto deben centrarse en la interfaz de usuario , independientemente de que su naturaleza sea de usar y tirar o evolutiva.

Las actuales herramientas de desarrollo rápido de aplicaciones (RAD, Rapid Application Development) permiten la construcción rápida de interfaces de usuario.

### **Productos entregables**

- ✓ Prototipo del sistema
- ✓ Conflictos detectados

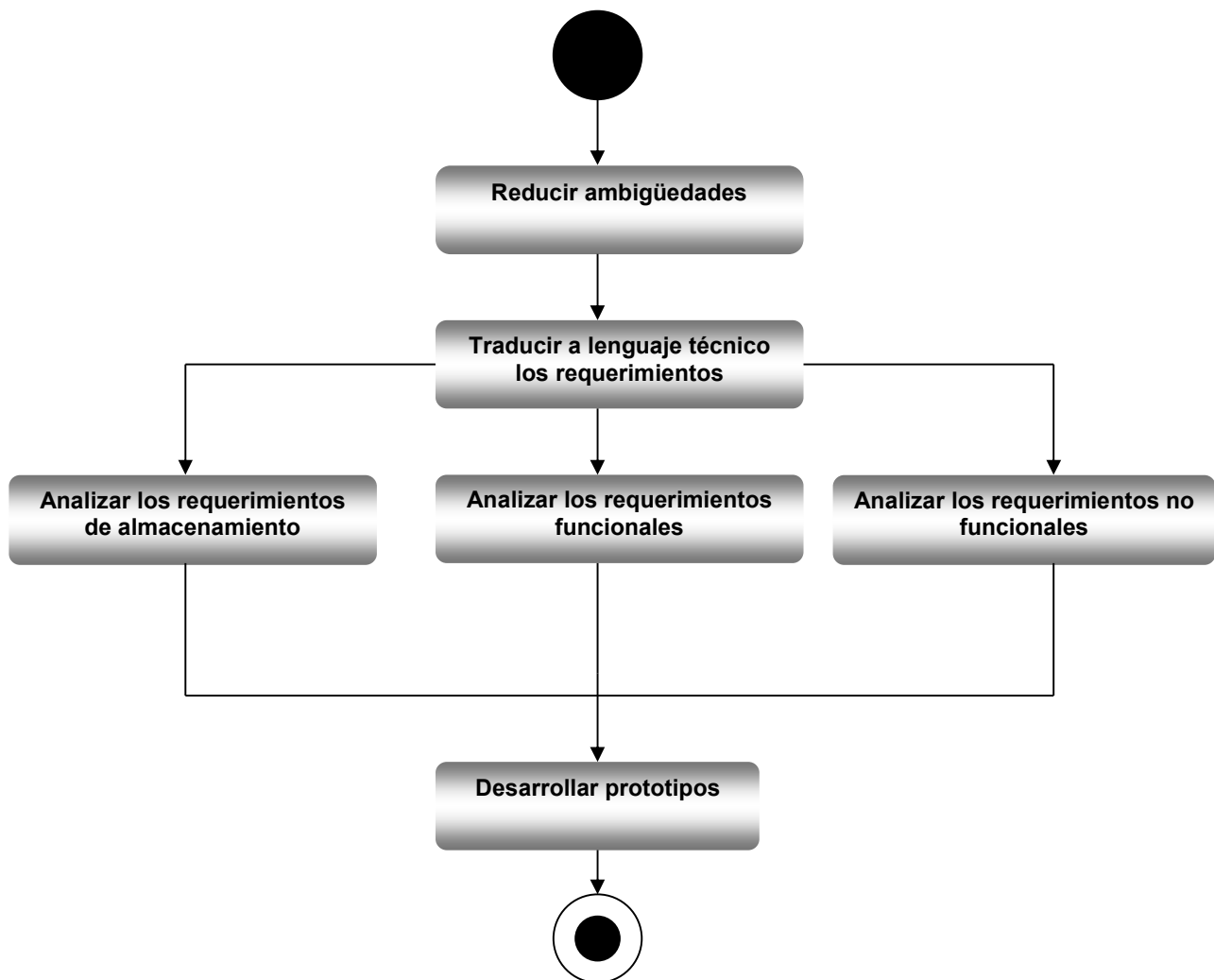
### **Técnicas**

- ✓ Prototipado de sistemas software

### **Herramientas**

- ✓ Herramientas de desarrollo rápido de aplicaciones (RAD, Rapid Application Development)

Si se hiciera una representación gráfica de la propuesta metodológica para la etapa el análisis de requerimientos, la misma quedaría bosquejada de la manera que se puede apreciar en la figura 4.3



**Figura 4.3: Diagrama de Actividades de la actividad de Análisis del Proceso de la Ingeniería de Requerimientos**

## **4.4 Documentación de los Requerimientos**

En esta fase se documentan los requerimientos acordados con el cliente, en un nivel apropiado de detalle. En la práctica, esta etapa se va realizando conjuntamente con el análisis, pero podríamos decir que la Documentación es el "pasar en limpio" el análisis realizado previamente aplicando técnicas y/o estándares de documentación, como la notación UML 2.0

### **4.4.1 Tarea 1: Documentar Etapa**

#### **Objetivos**

- ✓ Obtener el Documento de Requerimientos en su etapa inicial
- ✓ Integrar los diagramas, plantillas y todas las especificaciones de los requerimientos a un detalle apropiado.

#### **Descripción**

Esta etapa es de integración de la documentación que se ha generado en las actividades de elicitación y análisis, aquí se reunirán todos los productos entregables de las actividades anteriores según la estructura del documento de requerimientos, este es un documento formal con requerimientos analizados por los propios clientes, usuarios y desarrolladores que pasaran a la etapa de validación.

El objetivo de esta tarea es obtener un Documento de requerimientos que puedan utilizarse durante la validación de los requerimientos.

#### **Productos entregables**

- ✓ Documento de requerimientos del Sistema

#### **Técnicas**

- ✓ Recopilación de Documentos y Modelos

## **4.5 Documento de Requerimientos**

El documento de requerimientos es un documento formal usado para comunicar los requerimientos a clientes, ingenieros y administradores.

El documento de requerimientos describe:

Los servicios y funciones que el sistema deberá proveer.

- Las restricciones bajo las cuales el sistema deberá operar
- Las propiedades generales del sistema (i.e. restricciones en las propiedades “sobresalientes” del sistema)
- Definiciones de otros sistemas con los cuales el sistema deberá integrarse.
- Información sobre el dominio de la aplicación.
- Restricciones en el proceso de desarrollo.
- Restricciones en equipo (hardware) en que correrá el sistema

### **4.5.1 Usuarios del Documento de Requerimientos**

Clientes del Sistema: Especifican los requerimientos y los leen para verificar que cumplen con sus necesidades.

Administradores del Proyecto: Usan el documento para planear un costo del sistema y el proceso de desarrollo del mismo.

Ingenieros de sistemas: Usa el documento para entender el sistema siendo desarrollado.

Ingenieros de Pruebas del sistema Usa el documento para desarrollar pruebas de validación del sistema.

Ingenieros del mantenimiento del sistema: Use el documento para entender el sistema.

### **4.5.2 Estructura del Documento de Requerimientos**

IEEE/ANSI 830-1993 es el estándar propuesto para organizar la estructura del documento de requerimientos.

- Introducción

- 1.1 Propósito del Documento de Requerimientos
  - 1.2 Alcance del producto
  - 1.3 Definiciones, acrónimos y abreviaciones
  - 1.4 Referencias
  - 1.5 Panorama del resto del documento
  - Descripción General
    - 2.1 Perspectiva del producto
    - 2.2 Funciones del producto
    - 2.3 Características del usuario
    - 2.4 Restricciones generales
    - 2.5 Supuestos y dependencias.
  - Requerimientos específicos
    - 4.1 Cobertura de requerimientos funcionales, no-funcionales y de interfaz.
  - Apéndices
- Índice

El estándar de la IEEE es genérico con la intención de ser aplicado a una amplia variedad de documentos de requerimientos.

En general, no todas las partes del estándar son requeridas para los documentos de requerimientos.

Cada organización debería adaptar el estándar dependiendo de los sistemas que desarrolla.

#### **4.5.3 Definición de Partes del Documento del Requerimientos**

**Introducción:** La introducción debe proporcionar una apreciación global del documento completo.

Debe contener las subdivisiones siguientes:

**Propósito:**

Esta subdivisión debe:

- a) Delinear el propósito del documento;
- b) Especifique a que público intencional va dirigido el documento.

**Alcance:**

Esta subdivisión debe:

- a) Identificar el producto (s) del software para ser diseñado por el nombre (por ejemplo, Anfitrión DBMS, el Generador del Reporte, etc.);
- b) Explicar que hará y que no hará el producto de software.
- c) Describir la aplicación del software especificándose los beneficios pertinentes, objetivos, y metas;
- d) Mostrar consistencia con las declaraciones similares en las especificaciones de niveles superiores (por ejemplo, las especificaciones de los requerimientos del sistema), si ellos existen.

**Definiciones, siglas, y abreviaciones**

Esta subdivisión debe proporcionar las definiciones de todas las condiciones, las siglas, y abreviaciones que exigen interpretar el documento propiamente.

**Referencias**

Esta subdivisión debe:

- a) Proporcionar una lista completa de todas las referencias de los documentos en otras partes en el documento de requerimientos.
- b) Identificar cada documento por el título, número del reporte (si es aplicable), fecha, y publicación de la organización;
- c) Especificar las fuentes de las referencias de donde se obtuvieron.

**Resumen**

Esta subdivisión debe:

- a) Describa lo que el resto del documento de requerimientos contiene;

- b) Explica cómo el documento de requerimientos es organizado.

**Descripción global:** Esta sección debe describir los factores generales que afectan el producto y sus requerimientos. Esta sección no declara los requerimientos específicos. En cambio, mantiene un fondo de esos requerimientos que se definen en detalle y les hacen más fácil entender.

Esta sección normalmente consiste en seis subdivisiones, como sigue:

**Perspectiva del producto**

Esta subdivisión del SRS debe poner el producto en la perspectiva con otros productos relacionados. Si el producto es independiente y totalmente autónomo, debe declararse que así es. Si el documento de requerimientos define un producto que es un componente de un sistema más grande, como frecuentemente ocurre, entonces esta subdivisión debe relacionar los requerimientos de ese sistema más grande a la funcionalidad del software y debe identificar las interfaces entre ese sistema y el software.

Esta subdivisión también debe describir cómo el software opera dentro de las varias restricciones. Por ejemplo, estas restricciones podrían incluir:

- a) las interfaces del Sistema;
- b) las interfaces del Usuario;
- c) las interfaces del Hardware;
- d) las interfaces del Software;
- e) las interfaces de Comunicaciones;
- f) la Memoria;
- g) los Funcionamientos;
- h) los requerimientos de adaptación del Site.

**Funciones del Producto**

Esta subdivisión debe proporcionar un resumen de las funciones mayores que el software realizará.

Por ejemplo, documento de requerimientos para un programa de contabilidad puede acostumbrar esta parte a dirigirse al mantenimiento de Cuenta de

Cliente, declaración del cliente y preparación de la factura sin mencionar la inmensa cantidad de detalle que cada uno de esas funciones requiere.

a) Las funciones deben organizarse en cierto modo eso hace la lista de funciones entendible al cliente o a cualquiera nada más leyendo el documento la primera vez.

b) Pueden usarse los métodos Textuales o gráficos para mostrar las funciones diferentes y sus relaciones.

No se piensa que el diagrama muestra un diseño de un producto, sino simplemente muestra la relación lógica entre las variables.

### **Características del usuario**

Esta subdivisión debe describir esas características generales de los usuarios intencionales del producto que incluye nivel educativo, experiencia, y la especialización técnica.

### **Restricciones**

Esta subdivisión debe proporcionar una descripción general de cualquier otro punto que limitará las opciones de los diseñadores. Éstos incluyen:

- a) las políticas reguladoras;
- b) las limitaciones del Hardware;
- c) las Interfaces a otras aplicaciones;
- d) el funcionamiento Paralelo;
- e) las funciones de la Auditoría;
- f) las funciones de Control;
- g) los requerimientos de lenguaje;
- h) los protocolos Señalados (por ejemplo, XON-XOFF, ACK-NACK);
- i) los requerimientos de Fiabilidad;
- j) Credibilidad de la aplicación;
- k) la Seguridad y consideraciones de seguridad.

### **Atenciones y dependencias**

Esta subdivisión debe listar cada uno de los factores que afectan los requerimientos declarados en el documento de requerimientos.



Estos factores no son las restricciones del diseño en el software pero son, más bien, cualquier cambio a ellos eso puede afectar los requerimientos en el documento de requerimientos. Por ejemplo, una suposición puede ser que un sistema operativo específico estará disponible en el hardware designado para el producto del software. Si, de hecho, el sistema operativo no está disponible, los SRS tendrían que cambiar de acuerdo con entonces.

### **Requerimientos específicos**

Esta sección debe contener todos los requerimientos del software a un nivel de detalle suficiente para permitirles a los diseñadores diseñar un sistema para satisfacer esos requerimientos, y a los auditores a probar que el sistema satisface esos requerimientos. A lo largo de esta sección, cada requisito declarado debe ser externamente perceptible por los

usuarios, operadores u otros sistemas externos. Estos requerimientos deben incluir por lo menos una descripción de cada entrada (el estímulo) en el sistema, cada salida (la contestación) del sistema, y todas las funciones realizadas por el sistema en la salida a una entrada o en el apoyo de la salida. Esta es la parte más grande y más importante del documento de requerimiento, los principios siguientes aplican:

- a) Deben declararse los requerimientos específicos en la conformidad con todas las características descritas.
- b) Todos los requerimientos deben ser singularmente identificables.
- c) Debe prestarse la atención debida a organizar los requerimientos para aumentar al máximo la legibilidad.

### **Apéndices**

Los apéndices no siempre son considerados parte del SRS real y no siempre son necesarios. Ellos pueden incluir:

- a) Ejemplos de formatos de las entradas/salidas, las descripciones del análisis del costo que se estudiaron o resultados de estudios del usuario;
- b) Apoyando o dando información a fondo que puede ayudar a los lectores del documento de requerimientos;
- c) Una descripción de los problemas a ser resuelto por el software;

## **4.6 Técnicas**

A continuación, se describen algunas de las técnicas que se proponen en esta metodología para obtener los productos de las tareas que se han descrito.

### **4.6.1 Entrevistas y cuestionarios**

Las entrevistas y cuestionarios se emplean para reunir información proveniente de personas o grupos, información que se obtiene conversando con el encuestado.

Para la realización de las entrevistas debemos coordinar previamente la fecha y hora, y debemos realizar un plan de agenda, en el cual hacemos un punteo del objetivo de la entrevista. Por ejemplo, en la primer entrevista estableceremos un plan de comunicación, en el cual se intercambiarán los teléfonos, celulares, direcciones de e-mail y horarios de contacto.

Para realizar las entrevistas, conviene llevar preparado un cuestionario. En términos generales, un cuestionario consiste en un conjunto de preguntas presentadas a una persona para su respuesta. La forma de la pregunta puede influir en las respuestas, por lo que hay que planearlas cuidadosamente.

Las preguntas suelen distinguirse en dos categorías: abiertas y cerradas. Las preguntas abiertas permiten que los encuestados respondan con su propia terminología. Generalmente estas son más reveladoras, ya que los interrogados no están limitados en sus respuestas. Son especialmente útiles en la etapa exploratoria de la investigación, cuando el analista busca penetrar en el pensamiento del encuestado.

A continuación se presentan algunos ejemplos de preguntas abiertas:

- ¿Cuál es la razón por la que quiere resolver este problema?
- ¿Cómo se resuelve el problema actualmente?
- ¿Cuál es el valor de una solución exitosa?
- ¿Quién usará el sistema que se va a construir?
- ¿Cómo desearía llevar a cabo esta actividad?

Por su parte, las preguntas cerradas predeterminan todas las posibles respuestas y el interrogado elige entre las opciones presentadas.

Estas preguntas las podemos utilizar cuando estamos estableciendo el criterio de priorización de los casos de uso con el cliente. Para cada uno preguntamos si es a corto plazo, a futuro, o Indispensable. Como vemos, la respuesta está acotada a tres opciones. También podemos volver a utilizarla cuando tenemos que negociar algún requerimiento con el cliente.

### **Barreras de la Comunicación**

- OIR LO QUE QUEREMOS
- PASAR POR ALTO IDEAS CONTRARIAS
- DIFERENTE SIGNIFICADO DE LAS PALABRAS
- COMUNICACION NO VERBAL
- EMOCIONES
- RUIDO
- DISTANCIA

### **Eliminación de Barreras**

- ADAPTARSE AL MUNDO DEL RECEPTOR
- UTILIZAR EL DIALOGO
- SERVIRSE DE LA COMUNICACION DIRECTA
- INSISTIR (VARIAS VECES)
- UTILIZAR LENGUAJE SENCILLO Y DIRECTO
- Utilizar VIAS DISTINTAS
- REDUCIR LAS DISTANCIAS

### **Cualidades del Entrevistador**

- ☐ Saber observar y escuchar (escucha activa)
- ☐ Poseer madurez
- ☐ Ser objetivo e imparcial
- ☐ No ser autoritario
- ☐ Capacidad de “*empatía*”
- ☐ Comprensión
- ☐ Ser cordial y accesible

- ☐ Respetar la intimidad
- ☐ Ser sincero, paciente, sereno
- ☐ Ser prudente

#### **4.6.2 Sistemas existentes**

Esta técnica consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Por un lado, podemos analizar las interfases de usuario, observando el tipo de información que se maneja y cómo es manejada. Esto puede ser útil para descubrir información importante a tener en cuenta, información que tal vez el cliente/usuario haya fallado en comunicar. También es útil analizar las distintas salidas que los sistemas producen (listados, consultas, etc.), porque siempre pueden surgir nuevas ideas sobre la base de estas. Luego de este análisis también podemos realizar una tormenta de ideas y así obtener requerimientos sumamente interesantes.

Esta técnica se puede realizar a priori aunque con cierto grado de análisis e investigación sin que intervenga el cliente/usuario; para ello, existen en internet cantidad de demos de productos que pueden resultar similares y, también, podemos establecer contacto con profesionales que desarrollan sistemas de características comparables.

Otra ventaja que presenta esta técnica es que como estos sistemas ya están en producción, ya han pasado por la curva de aprendizaje del dominio del problema. Entonces, cuando analizamos estos sistemas, tenemos que tratar de pensar, por ejemplo, por qué manejan cierta información y para qué sirve, lo que resultará de suma utilidad para nuestro cliente.

También es recomendable que luego de haber analizado el sistema, se lo mostremos al cliente/usuario, ya que por su experiencia puede sugerir importantes ideas nuevas.

#### **4.6.3 Grabaciones de video y de audio**

Básicamente existen dos formas de utilizar las grabaciones: como registro y apoyo de las entrevistas, y para analizar algún proceso en particular.

En cuanto a su función de apoyo, es importante por cuanto permite centrar la atención en la entrevista en sí en vez de distraerse tomando notas de todo lo que se dice. Cuando se está grabando la conversación, basta con "puntear" en una libreta los temas tratados para después tener una guía básica de los temas tratados y saber en qué lugar de la grabación buscar. Además, permite analizar los temas con más detenimiento y con una visión más global, pues ya se ha conversado sobre todos los puntos necesarios y se han visto los procesos ad hoc.

Cuando se trata de analizar algún proceso en particular, su ayuda es inestimable (sobretudo las filmaciones de video) ya que permite ver y analizar en detalle ese proceso la cantidad de veces que sea necesario. Y no olvidemos que filmando el lugar de trabajo estamos capturando el proceso de trabajo, lo que evita que impongamos nuestras expectativas y preferencias.

Para finalizar, mencionamos que siempre es conveniente comenzar las grabaciones con preguntas poco importantes que sirvan para "relajar el ambiente", ya que el entrevistado puede ponerse nervioso durante los primeros minutos de grabación. Debemos darle tiempo a las personas para que se distiendan y se sientan cómodas con la idea de ser grabados o filmados.

#### **4.6.4 Brainstorming (tormenta de ideas)**

Este es un modelo que se usa para generar ideas. La intención en su aplicación es la de generar la máxima cantidad posible de requerimientos para el sistema.

No hay que detenerse en pensar si la idea es o no del todo utilizable. La intención de este ejercicio es generar, en una primera instancia, muchas ideas. Luego, se irán eliminando en base a distintos criterios como, por ejemplo, "caro", "impracticable", "imposible", etc.

Las reglas básicas a seguir son: los participantes deben pertenecer a distintas disciplinas y, preferentemente, deben tener mucha experiencia. Esto trae aparejado la obtención de una cantidad mayor de ideas creativas.

Conviene suspender el juicio crítico y se debe permitir la evolución de cada una de las ideas, porque sino se crea un ambiente hostil que no alienta la generación de ideas.

Por más locas o salvajes que parezcan algunas ideas, no se las debe descartar, porque luego de maduras probablemente se tornen en un requerimiento sumamente útil. A veces ocurre que una idea resulta en otra idea, y otras veces podemos relacionar varias ideas para generar una nueva. Escribir las ideas sin censura.

#### **4.6.5 Arqueología de documentos**

Con la aplicación de esta herramienta se tratan de determinar posibles requerimientos sobre la base de inspeccionar la documentación utilizada por la empresa; por ejemplo, boletas, facturas, remitos, etc.

Esta herramienta sirve más que nada como complemento de las demás técnicas, y nos ayuda a obtener información que de otra manera sería sumamente difícil conseguir. Por ejemplo, en las facturas podemos encontrar información que no se pensaba manejar y que en definitiva resulta de suma utilidad, como un número propio de la empresa que se utiliza para saber el orden que tiene la factura en la carpeta y que permite encontrar las copias del documento con mayor rapidez.

En definitiva, se debe recolectar cualquier formulario o documento que sea utilizado para registrar o enviar información.

Para el análisis de cada uno de estos documentos, debemos realizar algunas preguntas, como:

- ¿Cuál es el propósito de este documento?
- ¿Quién lo usa? ¿Por qué? ¿Para qué?
- ¿Cuáles son las tareas que realizan con este documento?
- ¿Se puede encontrar una relación entre los documentos?
- ¿Cuál es el proceso que realiza la conexión?
- ¿Cuál es el documento que da más problemas a los usuarios?

#### **4.6.6 Aprendiz**

Esta herramienta se basa en la idea del maestro y el aprendiz, y es una buena forma de observar el trabajo real. Aquí, el aprendiz es representado por el analista, y el usuario/cliente cumple el rol de maestro.

El aprendiz se sienta con el maestro a aprender por medio de la observación, haciendo preguntas como ¿por qué hizo eso? y ¿qué significa eso?, y también realizando algún trabajo bajo la supervisión del maestro.

Esta técnica puede ser combinada con la herramienta de modelo conceptual. A medida que el trabajo es observado y explicado, el analista puede realizar bosquejos para cada una de las tareas realizadas, y también puede bosquejar como se conectan por medio de los distintos flujos de datos.

La aplicación de esta herramienta es muy útil, ya que a veces es difícil para el cliente/usuario el explicar cómo realiza su trabajo. Es también una técnica apropiada para un proyecto donde el problema no es estructurado, ya que es una de las mejores formas de obtener el conocimiento que se encuentra en la "cabeza" del cliente.

Una de las posibles objeciones que se le pueden hacer a esta herramienta es que su implementación requiere de mucho tiempo.

#### **4.6.7 JAD (Joint Application Development/Desarrollo conjunto de aplicaciones):**

esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes (IBM, 1997). Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la aplicación de la técnica se trabajará sobre lo que se generará. Tras una fase de preparación del JAD al caso concreto, el equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requerimientos de alto nivel a trabajar, el ámbito del problema y la documentación.

Durante la sesión se discute en grupo sobre estos temas, llegándose a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema.

Esta técnica presenta una serie de ventajas frente a las entrevistas tradicionales, ya que ahorra tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado, pero requiere un grupo de participantes bien integrados y organizados.

#### **4.6.8 Etnografía**

Es sumamente difícil describir cómo hacer el nudo de un calzado deportivo, pero es sumamente fácil mostrar los pasos para hacerlo.

Observar como se hacen las cosas es una buena manera de entender lo que estas requieren. Conectarse íntimamente con la cultura de la organización, vivirla, es una herramienta que debe ser tomada en cuenta.

También podemos realizar filmaciones del lugar de trabajo, para luego observarlas y analizarlas, buscando patrones, procesos, problemas, etc.

Siempre tenemos que estar atentos a lo que sucede en el entorno de la organización; por ejemplo, ver cómo resuelven un problema que surge, como un llamado telefónico que puede ocurrir mientras estamos presentes.

Dentro de la estrategia de observar tenemos que tratar de buscar estructuras y patrones. La estructura del trabajo para los usuarios suele ser invisible, por lo que será nuestro trabajo realizar las abstracciones necesarias.

#### **4.6.9 Run Use Case WorkShop (Talleres de Trabajo basados en los Casos de Uso)**

Estos talleres de trabajo se realizan entre el cliente/usuario y el equipo de requerimientos. La primera parte del WorkShop consiste en generar los escenarios. Para esto se necesita la información que tiene para brindar el usuario/cliente.

La idea es conversar por medio de los casos de uso y extraer de los usuarios las cosas esenciales que suceden cuando ocurre un evento determinado. Así, tratamos de definir la serie de usuarios y reconocer los pasos que se realizan para el caso de uso en estudio.



Luego preguntamos si los pasos registrados están bien o si hay que cambiarlos o mejorarlos. Como resultado de este proceso obtenemos un excelente bosquejo del caso de uso.

Una vez finalizada la etapa anterior, el equipo de requerimientos retorna a la oficina a especificar y deducir los requerimientos, a partir del conocimiento previamente adquirido.

#### **4.6.10 Prototipos**

Durante la actividad de extracción de requerimientos, puede ocurrir que algunos requerimientos no estén demasiado claros o que no estemos muy seguros de haber entendido correctamente los requerimientos obtenidos hasta el momento, todo lo cual puede llevar a un desarrollo no eficaz del sistema final.

Entonces, para validar los requerimientos hallados, se construyen prototipos.

Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiéndonos conseguir una importante retroalimentación en cuanto a si el sistema diseñado en base a los requerimientos recolectados le permite al usuario realizar su trabajo de manera eficiente y efectiva.

Los prototipos se pueden clasificar en:

##### **a-Prototipo evolutivo**

Por ejemplo, cuando el usuario no puede o no está dispuesto a articular sus requerimientos de ninguna forma, sólo se podrán determinar mediante un proceso de ensayo y error. Así se van realizando evoluciones sobre la base del mismo prototipo hasta determinar claramente los requerimientos.

##### **b- Prototipo Bosquejado**

Para realizar esta clase de prototipo nos apoyamos, por un lado, en el rol del analista de requerimientos, simulando las respuestas del sistema y realizando bosquejos de las interfases de usuario; y, por otro, el usuario, que es quien realiza las entradas

("utiliza el prototipo"). También podemos llevar el caso de uso y bosquejar la interfase de usuario y, mediante el diálogo, manejamos la interactividad entre el usuario y el sistema.

Una de las ventajas más importantes es el poco esfuerzo que realizamos para aplicar los cambios. Como desventaja podemos mencionar que si bien capturamos la idea, el usuario no percibe como será realmente el diálogo hombre-máquina, sobre todo si existe un requerimiento no funcional como el de usabilidad.

### **c-Prototipo Tangible/usable**

Los términos tangible y usable se refieren a desarrollar una aplicación (software) que el usuario pueda utilizar, es decir, con la cual pueda interactuar como si fuera la aplicación final.

Cualquiera sea la herramienta de software, que elijamos utilizar, para desarrollar el prototipo, debemos recordar los siguientes puntos:

- ❑ Debe demandar poco esfuerzo para realizar los cambios
- ❑ Debe poseer amplia flexibilidad para el manejo de las interfases de usuario.
- ❑ Debe consumir poco tiempo para generar un nuevo prototipo (maqueta).

Cabe remarcar que tenemos que dejar bien en claro al usuario/cliente que se trata de un prototipo y que no es el producto final; para explicar esta diferencia podemos hacer una analogía con las maquetas de los arquitectos.

El prototipo tangible/usable también resulta de suma utilidad cuando nos encontramos frente a un requerimiento no funcional de usabilidad, más precisamente, el de facilidad de uso.

Entre las desventajas más importantes de los prototipos que podemos mencionar, se encuentran:

- ❑ Costo de entrenamiento/capacitación en la herramienta
- ❑ Costo de realizar el prototipo.
- ❑ Problema de calendario.
- ❑ Incompletitud (puede confundir a los usuarios, haciéndolos pensar que el producto final quedará como el prototipo, incompleto).

#### **4.6.11 Análisis FODA**

Con este análisis se intentan identificar las principales fortalezas, oportunidades, debilidades y amenazas con las que se enfrenta una empresa.

Entonces, por un lado, tenemos las oportunidades y las amenazas, que se refieren a los factores externos que pueden afectar el futuro del negocio.

Por otro lado, se encuentran las fuerzas y debilidades que son factores internos; estas fuerzas señalan ciertas estrategias cuya aplicación podría conducir al éxito, mientras que las debilidades señalan aquello que la empresa debe corregir.

Esta herramienta es sumamente útil para analizar la situación de una empresa y ver de qué forma podemos ayudar a disminuir las debilidades y amenazas, y cómo podemos aprovechar las oportunidades o cómo podemos crear nuevas oportunidades y cómo hacer aún más fuertes las fortalezas. También nos es útil para analizar el impacto de la solución planteada en cada uno de los cuadros.

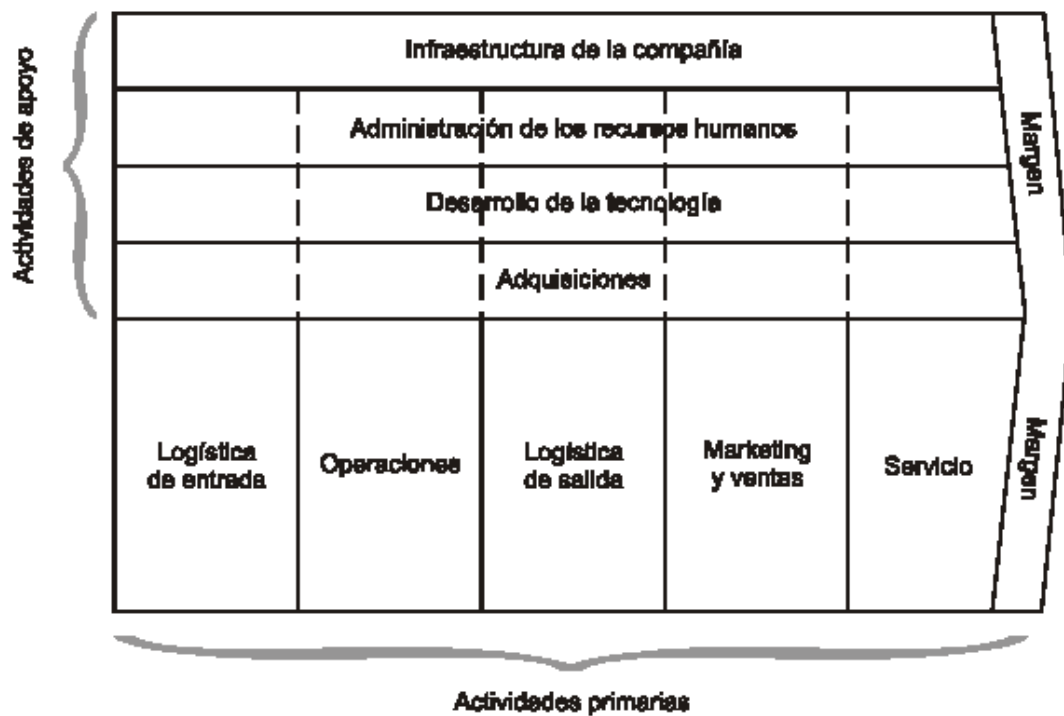
#### **4.6.12 Cadena de valor**

Todas las empresas son una colección de actividades que se llevan a cabo para diseñar, producir, distribuir, entregar y apoyar a su producto. La cadena de valor divide las empresas en nueve actividades estratégicas a fin de comprender el comportamiento de los costos en determinado negocio o industria y en las fuentes de diferenciación presentes y futuras.

En este análisis se deben examinar los costos y el funcionamiento de cada una de las actividades productoras de valor, tratando de mejorarlos.

Con esta herramienta podemos analizar los flujos de información que intervienen en las distintas actividades. Sirve también, por ejemplo, en el caso de que existan sistemas operacionales automatizados, para investigar qué tipo de información

maneja y para qué se utiliza. Analizando este modelo, podemos buscar la forma en la cual el sistema puede ayudar a obtener mayor valor para la actividad o conjunto de actividades estudiadas.



#### 4.6.13 Modelo de clase conceptual | Diagrama Conceptual | Diagrama de Clases Conceptual

Un modelo conceptual es una representación de conceptos del dominio del problema. [Fowler96]. Permite mostrar conceptos, asociaciones entre conceptos y atributos de conceptos. La creación del modelo también ayuda a comprender la terminología del dominio y comunica cuáles son los términos importantes y las relaciones existentes entre ellos.

Concepto: categoría de idea o cosas. La intención del concepto es la descripción de sus atributos, operaciones y significado. [Craig, L. 1999]

Para representar un concepto podemos basarnos en la metodología orientada a objetos, utilizando las clases como forma de representar el concepto, dado que una

clase representa un concepto del dominio del problema que abarca un amplio espectro, como un pedido, una máquina, una tarea, un trabajo, etc.

Esta herramienta puede ser utilizada para capturar el vocabulario del sistema que se está desarrollando. Mediante ella podemos incluir abstracciones que forman parte del dominio.

Para especificar el modelo conceptual utilizamos el diagrama de clases en el cual mostramos las relaciones existentes entre las clases. Es decir, mostramos los conceptos que se manejan en el negocio y como se relacionan entre sí.

Es una buena forma para obtener una idea general de cómo funciona el negocio (relaciones entre las clases/concepto), y capturar vocabulario y conceptos (clase). También es de ayuda para incluir nuevos conceptos al glosario.

#### **4.6.14 Diagrama de pescado (Ishikawa Diagram, Cause-and-Effect o Fishbone Diagram)**

Es una antigua pero útil herramienta que sirve, en el proceso de IR, para analizar problemas y comprender cuáles son sus causas. Por ejemplo, en el proyecto citado en el Anexo 1, se tenía que analizar el costo de los distintos componentes, en los cuales intervienen procesos, tecnología, gente.

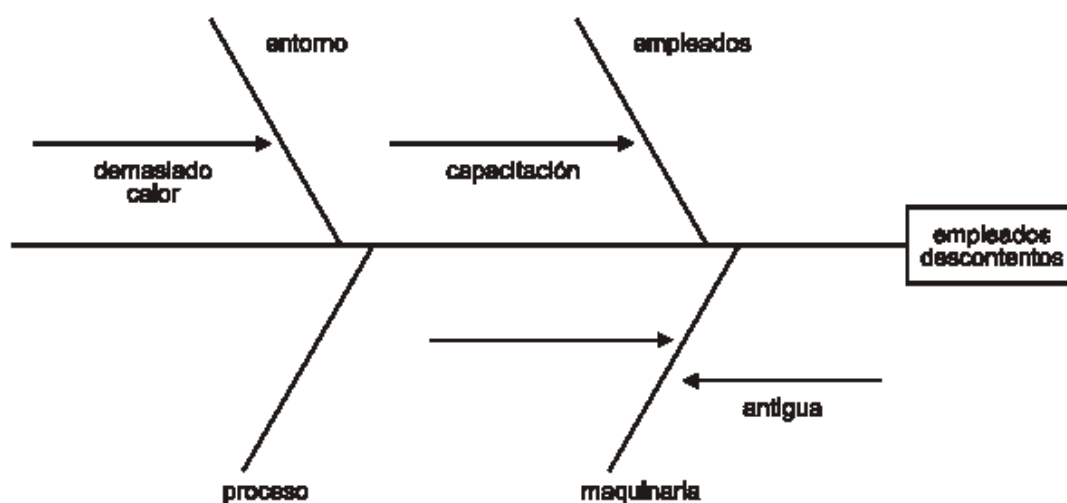
En la figura que se presenta como ejemplo del diagrama de pescado, se analizan las posibles causas por la que los empleados no están contentos con el trabajo que realizan.

Como vemos en este diagrama, se distinguen 4 categorías básicas: empleado, entorno, proceso, maquinaria (tengamos en cuenta que la aparición de estas categorías va a depender del caso que se esté estudiando). En dicha figura vemos causas relacionadas con los trabajadores, con el entorno, la maquinaria y el proceso en sí. Por ejemplo, se analiza una posible causa relacionada con la capacitación de los empleados, y otra que es el estado antiguo de la maquinaria de trabajo.

Con esta herramienta podemos analizar cómo impacta la solución planteada para un requerimiento dado. Por ejemplo, cómo afecta al trabajo diario del empleado.

También en el caso de que la solución planteada interactúe con otros sistemas existentes (modificando, consultando o intercambiando información) el diagrama de pescado nos permite analizar los posibles problemas que pueden surgir.

Esta herramienta la podemos utilizar conjuntamente con una tormenta de ideas (brainstorming), para ayudarnos a ordenar las posibles soluciones a un problema. Es decir, por un lado generamos ideas y luego utilizamos esta herramienta para organizarlas.



#### **4.6.15 Glosario**

El glosario es una simple lista de términos en donde se explica su significado. En esta lista se incluyen y definen todos los términos que requieren explicación, mejorando así la comunicación intergrupal y la comunicación con el cliente, y mitigando el riesgo de malos entendidos.

Los términos que se incluyen provienen de todas las áreas del proyecto: casos de uso, terminología propia del negocio, etc.

El glosario se va actualizando durante el transcurso del proceso de IR, perfeccionándolo en cada nuevo ciclo.

¿Cómo hacerlo?

Para realizar el glosario utilizamos dos columnas; en la primera ingresamos el nombre del término y, en la segunda, ingresamos su descripción.

Término	Descripción
---------	-------------

Es importante ordenar alfabéticamente esta tabla por Término, para así facilitar las consultas.

#### **4.6.16 Diagrama de Actividad**

Para representar un proceso de negocio podemos utilizar otra de las herramientas que nos proporciona el UML, que es el diagrama de actividad. El diagrama de actividad o diagrama de proceso, se asemeja a un mapa de procedimientos, mostrando el flujo de actividades: se toman decisiones (bifurcaciones) de acuerdo a las condiciones (condición de guarda), para luego pasar a la siguiente actividad o estado (transición). Este modelo también permite representar actividades que ocurren en paralelo, o aquellos casos en los que una única actividad desencadena más de una tarea (división de control), o cuando se unen dos o más actividades para formar una tercera (unión de control).

#### **4.6.17 Casos de uso**

El caso de uso es un documento narrativo que describe la secuencia de eventos de un actor (agente externo) que utiliza un sistema para completar un proceso. [Jacobson92]. Es una técnica diseñada para especificar el comportamiento de un sistema.

Este documento describe la posible secuencia de interacciones entre el sistema y uno o más actores, en respuesta a un estímulo inicial proveniente de un actor. No es una simple historia específica de eventos específicos intercambiados entre el sistema y los actores o escenario, sino que es una descripción de un conjunto de escenarios, cada uno de ellos comenzado con un evento inicial desde un actor hacia el sistema.

Los requerimientos se pueden expresar de diferentes formas, desde texto sin formato estricto hasta expresiones en un lenguaje formal, pasando por todas las formas intermedias. La mayoría de los requerimientos funcionales, sino todos, se pueden expresar con casos de uso.

#### **4.6.18 Casa de calidad QFD: Requerimientos versus Casos de Uso**

El esquema QFD (Quality Function Deployment) es una matriz que representa las casas de calidad, en las cuales las filas representan los "qué", o sea, la lista de los requerimientos, mientras que las columnas representan los "cómo", es decir, cómo se llevan a cabo los requerimientos (casos de uso).

Dado un requerimiento, se marcan todos los casos de uso que lo implementan y, dado un caso de uso, se marcan todos los requerimientos en los que éste participa.

Debemos recordar que todo requerimiento debe ser implementado a través de algún caso de uso y, que todo caso de uso debe satisfacer algún requerimiento.

#### **¿Cómo hacerlo?**

Para realizar esta técnica podemos utilizar un planilla de cálculo, como EXCEL®. Aquí se ingresará, en la filas, los identificadores de los requerimientos correspondientes a la lista de requerimientos. Los identificadores están compuestos por una letra y un número, por ejemplo, R1.

En tanto, en las columnas ingresamos un código que corresponde al identificador utilizado para cada uno de los casos de uso. Usualmente su formato está compuesto por letras y un número; por ejemplo, CU1.

Luego tomamos un requerimiento, por ejemplo el R1, el cual está implementado por los casos de uso CU1 y CU3; entonces buscamos la intersección y marcamos con una cruz, como se ve en la planilla de ejemplo.

No olvidemos que tenemos que verificar que todo requerimiento sea implementado por algún caso de uso y, que todo caso de uso satisface algún requerimiento.

Por otra parte, también tenemos que verificar la lista de requerimientos y la lista de casos de uso, revisando que no existan cosas repetidas redactadas de diferente manera.



#### **4.6.19 Checklist (lista de verificación)**

Esta herramienta es muy fácil de utilizar y proporciona una gran utilidad. En general es una lista de preguntas que el analista de requerimientos debe usar para evaluar cada requerimiento. Los analistas de requerimientos deben verificar y marcar los puntos de esta lista mientras leen el documento de requerimientos. Cuando se descubren problemas potenciales, deben ser anotados, ya sea en los márgenes del documento, ya sea en una lista de análisis.

Las checklist son útiles porque brindan un recordatorio de lo que se debe buscar y reducen las chances de pasar por alto alguna verificación importante. Y no sólo son útiles para verificar requerimientos; también se puede aplicar con los casos de uso y con los puntos a ser tratados en el plan de agenda.

##### **¿Cómo hacerlo?**

Este análisis se puede implementar con una hoja de cálculo en donde las filas representan, por ejemplo, los distintos requerimientos, y las columnas representan los puntos a verificar (el contenido de la checklist). Entonces, se completan las intersecciones que correspondan con los comentarios sobre los problemas potenciales.

#### **4.6.20 Análisis de Factores críticos de éxito**

Esta técnica consiste en realizar acciones necesarias para el cumplimiento de los factores claves de éxito y de esa manera se pueda alcanzar con éxito los objetivos del Proyecto. Los factores críticos de éxito más importantes son los siguientes: compromiso de los stakeholder, participación de los usuarios debido a su experiencia directa e indirecta, alcance claramente definido, es decir tener una idea clara de los objetivos, actividades y productos a obtener, y experiencia del grupo de trabajo.

#### **4.6.21 Análisis comparativo de las técnicas de ingeniería de requerimientos**

A continuación se presenta una tabla con las principales características de las técnicas y herramientas empleadas en la ingeniería de requerimientos.

Nombre	Clasificación	Actividad(es)	Ventajas	Desventajas
<b>Entrevistas</b>	Técnica	Extracción	Técnica conocida, fácil de aplicar, permite estudiar el dominio del problema, comunicación y retroalimentación con el usuario/cliente.	Requiere disposición del cliente.
<b>Cuestionarios</b>	Herramienta	Extracción	Herramienta fácil de aplicar, comprensible, varía según el tipo de preguntas aplicadas y su enfoque objetivo.	Están limitados Por el tipo de preguntas y el objetivo que se busca.
<b>Sistemas existentes</b>	Técnica	Extracción y Análisis	Útil para analizar las salidas de los sistemas actuales de la organización, permite descubrir requerimientos importantes, son sistemas que presentan estabilidad en el dominio del problema.	Diferencia en el enfoque y concepción del sistema, Necesidad (expresada por el usuario) de mejorar la aplicación actual.
<b>Grabaciones de video y de audio</b>	Herramienta	Extracción y Análisis	Permite centrar la atención en la entrevista, analizar los temas con más detenimiento, captura el proceso de trabajo.	Autorización del cliente.
<b>Brainstorming (Tormenta de ideas)</b>	Herramienta	Extracción y Análisis	Genera ideas, permite obtener varias vistas	Generación de múltiples ideas inconexas, inversión de tiempo

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

ideas)			del problema.	Tiempo en clasificación y eliminación de ideas.
Arqueología de documentos	Herramienta	Extracción y Análisis	Ayuda a determinar posibles requerimientos implícitos en los resultados o salidas de los sistemas de uso actual.	Bases incorrectas, las cuales se quieren mejorar y posiblemente no se tenga en cuenta este factor
Aprendiz	Técnica	Extracción	Permite observar el trabajo en tiempo real, realizar bosquejos para cada actividad a sistematizar, obtener la información que está en la cabeza del cliente.	Requiere mucho tiempo, Disposición del cliente.
Observación	Técnica	Extracción	Comprensión de los procesos y su entorno, permite identificar estructuras y patrones, facilita la abstracción.	Inversión grande de tiempo por parte del analista.
Run Use Case Workshop (talleres)	Herramienta	Extracción	Definir actores, escenarios, secuencia de actividades, eventos, retroalimentación continua del cliente.	Se necesita tener un bosquejo inicial de casos de uso.
Prototipo bosquejado	Herramienta	Extracción, Análisis y Especificación	Retroalimentación por parte del cliente/usuario, poco tiempo y esfuerzo en la aplicación de cambios.	El usuario no Alcanza a visualizar el diálogo hombre- máquina (requerimiento no funcional)
Prototipo tangible/ Usable	Herramienta	Extracción – Especificación y Validación	Permite visualizar los requerimientos no funcionales, retroalimentación, validación de entradas y salidas.	Costo de realización, entrenamiento, confusión de los usuarios por incompletitud.
DOFA	Herramienta	Análisis	Permite identificar las debilidades, oportunidades, fortalezas y amenazas del sistema.	
Cadena de valor	Herramienta	Análisis	Identificar las Actividades estratégicas de la empresa, análisis de costos y funcionamiento de cada actividad.	
Modelo de Clase conceptual	Herramienta	Análisis, especificación	Identificar conceptos del dominio del problema, comprensión de la terminología, facilita la construcción del glosario.	Es necesario contar con un Glosario del dominio del problema

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

<b>Diagrama de pescado</b>	Herramienta	Extracción – Análisis y Especificación	Impacto de las soluciones planteadas en la empresa, organización de ideas.	Dependiente del caso de estudio, interpretación sesgada.
<b>Glosario</b>	Herramienta	Extracción – Análisis – Especificación y Validación	Mejora la comunicación intergrupar, con el cliente, mitiga el riesgo de malos entendidos.	Requiere continua actualización.
<b>DCO</b>	Técnica	Análisis y Especificación	Comprensión del entorno del negocio, funcionamiento e interacción con el ambiente, permite modelar recursivamente la empresa en sus diferentes niveles.	
<b>Diagrama de actividad</b>	Herramienta	Análisis y Especificación	Permite representar las actividades, flujos de información, condiciones y restricciones.	Herramienta no apta para la comprensión del usuario, este requiere cierta capacitación.
<b>ESRE</b>	Técnica	Extracción – Análisis – Especificación y Validación	Especificar los requerimientos del sistema.	Sólo incluye requerimientos del producto.
<b>Casos de uso</b>	Herramienta	Extracción – Análisis – Especificación y Validación	Especifica el Funcionamiento del sistema, describe la secuencia de interacciones entre el sistema y los actores, descripción de un conjunto de escenarios.	
<b>Casa de calidad o QFD</b>	Herramienta	Validación	Representa la relación de qué (requerimiento) con el cómo (casos de uso), control de calidad.	
<b>Checklist</b>	Herramienta	Especificación y Validación	Evaluar cada requerimiento, recuerdan que se debe buscar y ajustar.	

Se presentan las principales ventajas y desventajas de algunas de las técnicas utilizadas en las etapas de la Ingeniería de Requerimientos.

<b>Técnica</b>	<b>Ventajas</b>	<b>Desventajas</b>
	<ul style="list-style-type: none"> <li>Mediante ellas se obtiene una gran cantidad de información correcta a través del usuario.</li> <li>Pueden ser usadas para obtener</li> </ul>	<ul style="list-style-type: none"> <li>La información obtenida al principio puede ser redundante o incompleta.</li> <li>Si el volumen de información</li> </ul>

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

<b>Entrevistas y Cuestionarios</b>	<p>un pantallazo del dominio del problema.</p> <ul style="list-style-type: none"> <li>• Son flexibles.</li> <li>• Permiten combinarse con otras técnicas.</li> </ul>	<p>manejado es alto, requiere mucha organización de parte del analista, así como la habilidad para tratar y comprender el comportamiento de todos los involucrados.</p>
<b>Lluvia de Ideas</b>	<ul style="list-style-type: none"> <li>• Los diferentes puntos de vista y las confusiones en cuanto a terminología, son aclaradas por expertos.</li> <li>• Ayuda a desarrollar ideas unificadas basadas en la experiencia de un experto.</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>• Es necesaria una buena compenetración del grupo participante.</li> </ul>
<b>Prototipos</b>	<ul style="list-style-type: none"> <li>• Ayudan a validar y desarrollar nuevos requerimientos.</li> <li>• Permite comprender aquellos requerimientos que no están muy claros y que son de alta volatilidad.</li> </ul>	<ul style="list-style-type: none"> <li>• El cliente puede llegar a pensar que el prototipo es una versión del software que será desarrollado.</li> <li>• A menudo, el desarrollador hace compromisos de implementación con el objetivo de acelerar la puesta en funcionamiento del prototipo</li> </ul>
<b>Análisis Jerárquico</b>	<ul style="list-style-type: none"> <li>• Permite determinar el grado de importancia de cada requerimiento.</li> <li>• Ayuda a identificar conflictos en los requerimientos.</li> <li>• Muestra el orden en que deben ser implementados los requerimientos.</li> </ul>	<ul style="list-style-type: none"> <li>• Debe construirse un estándar claro de evaluación, que incluya la participación del cliente.</li> </ul>
<b>Casos de Uso</b>	<ul style="list-style-type: none"> <li>• Representan los requerimientos desde el punto de vista del usuario.</li> <li>• Permiten representar más de un rol para cada afectado.</li> <li>• Identifica requerimientos estancados, dentro de un conjunto de requerimientos.</li> </ul>	<ul style="list-style-type: none"> <li>• En sistemas grandes, toma mucho tiempo definir todos los casos de uso.</li> <li>• El análisis de calidad depende de la calidad con que se haya hecho la descripción inicial.</li> </ul>

En base a las ventajas y desventajas mostradas anteriormente, se hace una comparación entre algunas de las técnicas.

**Entrevistas vs. Casos de Uso:** Un alto porcentaje de la información recolectada durante una entrevista, puede ser usada para construir casos de uso. Mediante esto, el equipo de desarrollo puede entender mejor el ambiente de trabajo de los involucrados. Cuando el analista sienta que tiene dificultades para entender una tarea, pueden recurrir al uso de un cuestionario y mostrar los detalles recavados en un caso de uso. De hecho, durante las entrevistas cualquier usuario puede utilizar diagramas de casos de uso para explicar su entorno de trabajo.

**Entrevistas vs. Lluvia de Ideas:** Muchas de las ideas planteadas en el grupo, provienen información recopilada en entrevistas o cuestionarios previos. Realmente la lluvia de ideas trata de encontrar las dificultades que existen para la comprensión de términos y conceptos por parte de los participantes; de esta forma se llega a un consenso.

**Casos de Uso vs. Lluvia de Ideas:** La lista de ideas proveniente del brainstorm puede ser representada gráficamente mediante casos de uso.

El siguiente cuadro muestra las técnicas que pueden ser utilizadas en las diferentes actividades de la IR.

	Elicitación	Análisis	Validación
Entrevistas y Cuestionarios	X		X
Lluvia de Ideas	X		X
Prototipos			X
Análisis Jerárquico	X		X
Casos de Uso	X	X	X

**Tabla 4.2 :Cuadro de Técnicas VS las diferentes actividades de la Ingeniería de Requerimientos**

#### **4.7 Plantillas utilizadas en la elicitación y análisis de requerimientos**

Las plantillas nos ayudan a comunicarnos con nuestros clientes y usuarios para el registro y gestión de requerimientos.

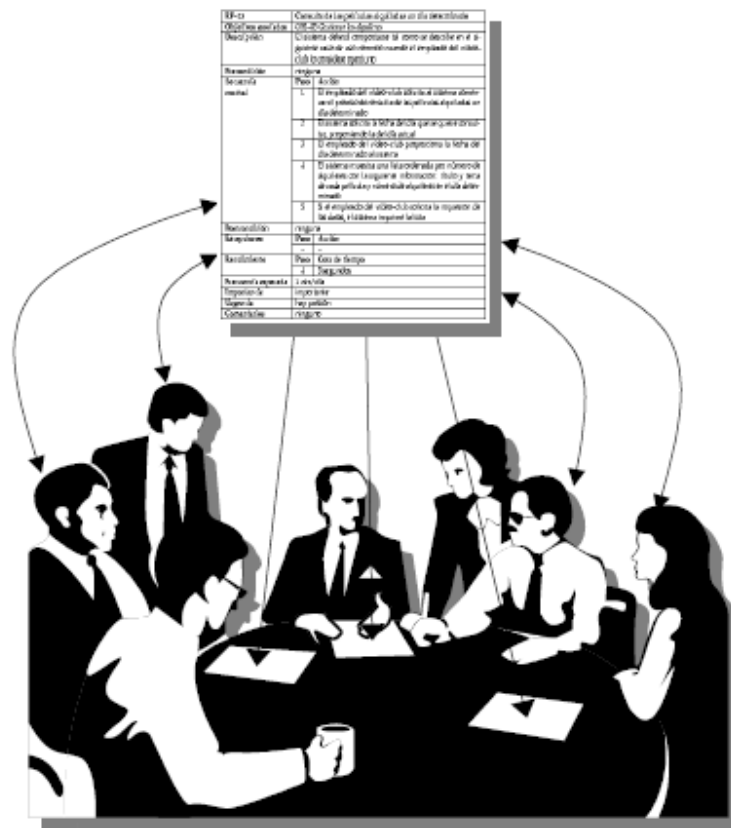
##### **Objetivo**

- Intentan paliar la falta de propuestas concretas sobre la expresión de requerimientos
- Se usan como elementos de elicitación y negociación durante las reuniones con clientes y usuarios facilitando la redacción de los requerimientos permitiendo a los participantes en las actividades de elicitación centrarse en expresar sus necesidades y no en cómo expresarlas.
- los participantes manejan directamente la documentación final, favoreciéndose así su implicación en el proceso.

##### **Notación Usada en las Plantillas**

En la notación usada en las plantillas propuestas se distinguen los siguientes patrones:

- las palabras o frases entre <Y> deben ser convenientemente reemplazadas.
- las palabras o frases que se encuentren entre {Y} y separadas por comas representan opciones de las que se debe escoger una.



**Figura 4.4: Las plantillas como elemento de elicitación y negociación**

En las siguientes secciones se describen las plantillas propuestas y la Notación anteriormente descrita.

#### **4.7.1 Plantilla para los objetivos del sistema**

Los objetivos del sistema pueden considerarse como requisitos de alto nivel, de forma que los requisitos propiamente dichos serían la forma de alcanzar los objetivos. A continuación la plantilla de objetivos del sistema:

<b>OBJ- &lt;id&gt;</b>	<b>&lt;nombre descriptivo&gt;</b>
<b>Versión</b>	<no de la versión actual> (<fecha de la versión actual>)
<b>Autores</b>	<autor de la versión actual> (<organización del autor>)
<b>Fuente</b>	<fuente de la versión actual> (<organización de la fuente>)
<b>Descripción</b>	<descripción del objetivo>
<b>Importancia</b>	<importancia del objetivo>
<b>Estado</b>	<estado del objetivo>



<b>Estabilidad</b>	<estabilidad del objetivo>
<b>Comentarios</b>	<comentarios adicionales sobre el objetivo>

Tabla 4.3: Plantilla de Objetivos del Sistema

### Significado de los campos

- 1. Identificador y nombre descriptivo:** Cada objetivo debe identificarse por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los objetivos comienzan con *OBJ*.
- 2. Versión:** Para poder gestionar distintas versiones, este campo contiene el número y la fecha de la versión actual del objetivo.
- 3. Autores:** Estos campo contienen el nombre y la organización de los autores (normalmente desarrolladores).
- 4. Fuente:** estos campos contienen el nombre y la organización de las fuentes (clientes o usuarios), de la versión actual del objetivo, de forma que la rastreabilidad pueda llegar hasta las personas que propusieron la necesidad del requisito.
- 5. Descripción:** Descripción del objetivo.
- 6. Importancia:** Este campo indica la importancia del cumplimiento del objetivo para los clientes y usuarios. Se puede asignar un valor o expresión numérico .En el caso de que no se haya establecido aún la importancia, se puede indicar que está por determinar.
- 7. Estado:** este campo indica el estado del objetivo desde el punto de vista de su desarrollo. El objetivo puede estar *en construcción* si se está elaborando, *pendiente de negociación* si tiene algún conflicto asociado pendiente de solución, *pendiente de validación* si no tiene ningún conflicto pendiente y está a la espera de validación o, por último, puede estar *validado* si ha sido validado por clientes y usuarios.
- 8. Estabilidad:** Este campo indica la estabilidad del objetivo, es decir una estimación de la probabilidad de que pueda sufrir cambios en el futuro. Esta estabilidad puede indicarse mediante un valor numérico o mediante una expresión enumerada como alta, media o baja o PD en el caso de que aún no se haya determinado.

**9. Comentarios:** Cualquier otra información sobre el objetivo que no encaje en los campos anteriores puede recogerse en este apartado.

#### 4.7.2 Plantilla para Requerimientos de almacenamiento de la información

Lo más importante en los sistemas de información es precisamente la información que gestionan. Las plantillas para requisitos de almacenamiento y de restricciones de información, que se verán ayudan a los clientes y usuarios a responder a las preguntas "¿qué información, relevante para los objetivos de su negocio, debe ser almacenada por el sistema? 2 "¿qué restricciones o reglas de negocio debe cumplir dicha información?".

<b>(*) &lt;nombre descriptivo&gt;</b>	<nombre descriptivo>
<b>Versión</b>	<no de la versión actual> (<fecha de la versión actual>)
<b>Autores</b>	<autor de la versión actual> (<organización del autor>)
<b>Fuente</b>	<fuente de la versión actual> (<organización de la fuente>)
<b>Descripción</b>	<descripción del requerimiento>
<b>Objetivos asociados</b>	<nombre del Objetivo>
<b>Requerimiento asociados</b>	<nombre del Requerimiento>
<b>Importancia</b>	<importancia del requerimiento>
<b>Estado</b>	<estado del requerimiento>
<b>Estabilidad</b>	<estabilidad del requerimiento>
<b>Comentarios</b>	<comentarios adicionales sobre el requerimiento>

Tabla 4.4: Plantilla para requisitos de almacenamiento de información

(\*)

**IRQ-<id>** : Requerimiento de Almacenamiento

**CRQ-<id>** : Requerimiento de Restricciones

### **Significado de los campos**

1. **Identificador y nombre descriptivo:** Cada requerimiento se debe identificar por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los requerimientos de almacenamiento de información comienzan con *IRQ* y los de requerimientos de restricciones de información con *CRQ*.
2. **Versión, Autores, Fuentes:** Estos campos tienen el mismo significado que en la plantilla para objetivos aunque referidos a los requerimientos de información y al de restricciones.
3. **Objetivos asociados:** Este campo debe contener una lista con los objetivos a los que está asociado el requisito, es decir de los objetivos de los que depende. Esto permite conocer qué requisitos harán que el sistema a desarrollar alcance los objetivos propuestos y justifican de esta forma la existencia o propósito del requisito.
4. **Requisitos asociados:** En este campo se indican otros requisitos que estén asociados por algún motivo con el requisito que se está describiendo, es decir de los requisitos de los que depende. De esta forma se posibilita tener una rastreabilidad *horizontal*.
5. **Descripción:** Describe al requerimiento.
6. **Importancia, Estado, Estabilidad, Comentarios:** Estos campos tienen el mismo significado que en la plantilla para objetivos aunque referidos al requisito.

#### **4.7.3 Plantilla para Requerimiento Funcionales**

Los sistemas de información no sólo almacenan información, también deben proporcionar servicios usando la información que almacenan. Las plantillas de requerimientos funcionales, que se verán a continuación, describen casos de uso y ayudan a los clientes y usuarios a responder a la pregunta "*¿qué debe hacer el sistema con la información almacenada para alcanzar los objetivos de su negocio?*".

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

<b>FRQ--&lt;id&gt;</b>	<nombre descriptivo>
<b>Versión</b>	<no de la versión actual> (<fecha de la versión actual>)
<b>Autores</b>	<autor de la versión actual> (<organización del autor>)
<b>Fuente</b>	<fuente de la versión actual> (<organización de la fuente>)
<b>Descripción</b>	<descripción del requerimiento>
<b>Objetivos Asociados</b>	<nombre del Objetivo>
<b>Requerimiento asociados</b>	<nombre del Requerimiento>
<b>Caso Uso Asociado</b>	<Caso Uso Asociado>
<b>Importancia</b>	<importancia del requerimiento>
<b>Estado</b>	<estado del requerimiento>
<b>Estabilidad</b>	<estabilidad del requerimiento>
<b>Comentarios</b>	<comentarios adicionales sobre el requerimiento>

**Tabla 4.5: Plantilla para Requerimientos Funcionales**

CU-<id>	NOMBRE DEL CASO DE USO	
DESCRIPCION	<descripción del caso de uso>	
ACTOR(ES)	<persona(s) involucradas en la ejecución del caso de uso>	
PRECONDICIONES	<precondición del caso de uso>	
POSTCONDICIONES	<poscondición del caso de uso>	
ESCENARIO PRIMARIO	<nombre del escenario principal a darse en el caso de uso>	
DESCRIPCIÓN DEL FLUJO		
Actor		Sistema
<acción del actor>		<acción del sistema>

EXCEPCIONES	<nombre de la excepción>	
<acción del actor>		<acción del sistema>
ESCENARIO SECUNDARIO	<nombre del escenario secundario a darse en el caso de uso>	
DESCRIPCIÓN DEL FLUJO		
Actor		Sistema
EXCEPCIONES	<nombre de la excepción>	
<acción del actor>		<acción del sistema>

**Tabla 4.6: Plantilla para Caso Uso**

### Significado de los campos

- 1. Identificador y nombre descriptivo:** Los identificadores de los requisitos funcionales empiezan con *FRQ* , si se desea solo resumir con algunas filas que conformen un Caso de Uso propiamente dicho, se reemplaza el FRQ con CU para que se encuentren referenciados. Generalmente el nombre descriptivo suele coincidir con el objetivo que los actores esperan alcanzar al realizarlo. No se debe confundir este objetivo con los objetivos del sistema. El objetivo que los actores esperan alcanzar al realizar un caso de uso es de más bajo nivel, por ejemplo *registrar un nuevo socio o consultar los pedidos pendientes*.
- 2. Descripción:** Se explica en forma breve y genérica al caso de uso al cual van hacer referencia.
- 3. Precondición:** En este campo se expresan en lenguaje natural las condiciones necesarias para que se pueda realizar el caso de uso. Estas condiciones se establecen bien sobre el entorno en el que opera el sistema, y que por lo tanto quedarán fuera de su control, bien sobre el estado del propio sistema.
- 4. Secuencia normal:** Este campo contiene la secuencia normal de interacciones
- 5. del caso de uso.** En cada paso, un actor o el sistema realiza una o más acciones, o se realiza (se *incluye*) otro caso de uso. Un paso puede tener una

condición de realización, en cuyo caso si se realizara otro caso de uso se tendría una relación de *extensión*. Se asume que, después de realizar el último paso, el caso de uso termina.

- 6. Postcondición:** En este campo se expresan en lenguaje natural las condiciones que se deben cumplir después de la terminación normal del caso de uso. Al igual que en el caso de las precondiciones, las postcondiciones se pueden establecer tanto sobre el entorno del sistema como sobre el estado del propio sistema.
- 7. Excepciones:** Este campo especifica el comportamiento del sistema en el caso de que se produzca alguna situación excepcional durante la realización de un paso determinado.
- 8.** Después de realizar las acciones o el caso de uso asociados a la excepción (una extensión), el caso de uso puede continuar la secuencia
- 9. Estabilidad:** Este campo indica la estabilidad del requerimiento, es decir una estimación de la probabilidad de que pueda sufrir cambios en el futuro. Esta estabilidad puede indicarse mediante un valor numérico o mediante una expresión enumerada como alta, media o baja o PD en el caso de que aún no se haya determinado.
- 10. Comentarios:** Cualquier otra información sobre el requerimiento que no encaje en los campos anteriores puede recogerse en este apartado.

#### **4.7.4 Plantilla para requerimiento No funcional**

Los requerimientos no funcionales del sistema se pueden expresar usando la plantilla siguiente:

<b>NFR-&lt;id&gt;</b>	<nombre descriptivo>
<b>Versión</b>	<no de la versión actual> (<fecha de la versión actual>)
<b>Autores</b>	<autor de la versión actual> (<organización del autor>)
<b>Fuentes</b>	<fuentes de la versión actual> (<organización de la fuente>)
<b>Descripción</b>	<descripción del requerimiento>
<b>Requerimiento asociados</b>	<nombre del Requerimiento>

<b>Importancia</b>	<i>&lt;importancia del Requerimiento &gt;</i>
<b>Estado</b>	<i>&lt;estado del requerimiento&gt;</i>
<b>Estabilidad</b>	<i>&lt;estabilidad del Requerimiento &gt;</i>
<b>Comentarios</b>	<i>&lt;comentarios adicionales sobre el Requerimiento &gt;</i>

**Tabla 4.7: Plantilla para requerimiento no funcional**

Los requisitos no funcionales del sistema se pueden expresar usando la plantilla que puede verse en la Tabla 4.7. El único campo específico de esta plantilla es la descripción, que debe completarse con **NFR** y la numeración correspondiente capacidad que deberá presentar el sistema, el significado del resto de los campos es el mismo que para las plantillas anteriores.

#### **4.7.5 Plantilla para conflictos**

Como ya se ha comentado, durante las sesiones de elicitación puede ser necesario resolver mediante algún tipo de negociación posibles conflictos en los requisitos elicitados en iteraciones previas del proceso. Para documentar dichos conflictos, y las soluciones adoptadas, se propone la plantilla siguiente:

<b>CFL--&lt;id&gt;</b>	<i>&lt;nombre descriptivo&gt;</i>
<b>Versión</b>	<i>&lt;no de la versión actual&gt; (&lt;fecha de la versión actual&gt;)</i>
<b>Autores</b>	<i>&lt;autor de la versión actual&gt; (&lt;organización del autor&gt;)</i>
<b>Fuentes</b>	<i>&lt;fuentes de la versión actual&gt; (&lt;organización de la fuente&gt;)</i>
<b>Objs./Reqs. en conflicto</b>	<i>&lt;nombre del objetivo o requisito en conflicto&gt;</i>
<b>Descripción</b>	<i>&lt;descripción del conflicto&gt;</i>
<b>Alternativas</b>	<i>&lt;alternativas propuestas&gt;</i>
<b>Solución</b>	<i>&lt;solución negociada&gt;</i>
<b>Requerimientos asociados</b>	<i>&lt;nombre del Requerimiento&gt;</i>
<b>Importancia</b>	<i>&lt;importancia de la resolución del conflicto&gt;</i>

<b>Estado</b>	<estado del requerimiento>
<b>Comentarios</b>	<comentarios adicionales sobre el conflicto >

**Tabla 4.8: Plantilla para conflictos**

### **Significado de los campos**

- 1. Identificador y nombre descriptivo:** Cada conflicto se debe identificar por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de conflictos comienzan con
- 2. Versión, Autores, Fuentes:** estos campos tienen el mismo significado que en las plantillas para objetivos y requerimiento, aunque referidos a los conflictos. En este caso especial, las fuentes son los participantes que deben participar en las posibles negociaciones necesarias para su resolución.
- 3. Objetivos y requisitos en conflicto:** este campo debe contener una lista con los objetivos y/o requisitos afectados por el conflicto.
- 4. Descripción:** este campo debe contener la descripción del conflicto.
- 5. Alternativas:** este campo debe contener una lista con las posibles alternativas de solución que se hayan identificado para solucionar el conflicto así como los autores de dicha alternativas.
- 6. Solución:** este campo debe contener la descripción de la solución negociada del conflicto, una vez que se haya acordado.
- 7. Requerimientos Asociados:** este campo debe contener una lista con los objetivos y/o requisitos afectados por la solución.
- 8. Importancia:** estos campos indican respectivamente la importancia de la resolución del conflicto.
- 9. Estado:** este campo indica el estado de resolución del conflicto, que podrá estar *en negociación* o bien *resuelto*.
- 10. Comentarios:** Estos campos tienen el mismo significado que en las plantillas anteriores.



#### 4.7.6 Plantilla para clases

<b>Clase</b>	<b>&lt;nombre clase&gt;</b>
<b>Versión</b>	<b>&lt;no de la versión actual&gt; (&lt;fecha de la versión actual&gt;)</b>
<b>Autores</b>	<b>&lt;autor de la versión actual&gt; (&lt;Organización del autor&gt;)</b>
<b>Fuente</b>	<b>&lt;fuente de la versión actual&gt; (&lt;Organización del fuente&gt;)</b>
<b>Requerimientos asociados</b>	<b>&lt;nombre del requerimiento&gt;</b>
<b>Descripción</b>	<b>&lt;descripción de la clase&gt;</b>
<b>Superclases</b>	<b>&lt;superclases directa de la clase&gt;</b>
<b>Subtipos</b>	<b>&lt;subclases directa de la clase&gt;</b>
<b>Comentarios</b>	<b>&lt;comentarios adicionales sobre la clase&gt;</b>

Tabla 4.9: Plantilla para Clases

#### Significado de los campos

- 1. Nombre Clase:** Cada clase se identifica por un nombre, que deberá ser único en el modelo y que se recomienda que sea un sustantivo o similar en singular.
- 2. Versión, Autores, Fuente:** estos campos tienen el mismo significado que en las plantillas para objetivos y requerimiento, aunque referidos a las clases.
- 3. Descripción:** En este apartado se indica si la clase es abstracto o concreto y se describe el concepto que representa.
- 4. Requisitos asociados:** En este apartado se indican los requisitos del *Documento de Requisitos del Sistema* a los que está asociado la clase, es decir, aquellos requisitos que justifican su existencia.
- 5. Supertipos:** En este apartado se indican los supertipos directos de la clase
- 6. Subtipos:** En este apartado se indican los subclases directos de la clase,
7. especificando si son disjuntos o solapados.
- 8. Comentarios:** Cualquier otra información sobre las clases.

#### 4.7.7 Plantilla para asociación

<b>Asociación</b>	<b>&lt;Clase1-Clase2 &gt;</b>
-------------------	-------------------------------

<b>Versión</b>	<i>&lt;no de la versión actual&gt; (&lt;fecha de la versión actual&gt;)</i>
<b>Autores</b>	<i>&lt;autor de la versión actual&gt; (&lt;Organización del autor&gt;)</i>
<b>Fuente</b>	<i>&lt;fuente de la versión actual&gt; (&lt;Organización del fuente&gt;)</i>
<b>Descripción</b>	<i>&lt;descripción de la asociación&gt;</i>
<b>Comentarios</b>	<i>&lt;comentarios adicionales sobre la clase&gt;</i>

**Tabla 4.10:      Plantilla de Asociación**

**Significado de los campos**

- 1. Asociación:** Se indica las clases asociadas..
- 2. Versión, Autores, Fuente:** estos campos tienen el mismo significado que en las plantillas para objetivos y requerimiento, aunque referidos a la asociación.
- 3. Descripción:** En este apartado se detalla la asociación.
- 4. Comentarios:** Cualquier otra información sobre la asociación

## **4.8 Caso Práctico: Sistema logístico de una Distribuidora de alimentos**

El Sistema que hemos tomado como Caso Practico es un Sistema Logístico, centrándonos en la parte de Almacén.

### **4.8.1 Utilizando la metodología propuesta para la actividad de Elicitación**

#### **1 Seleccionar el equipo de trabajo**

En esta tarea se identificó al personal capacitado para desarrollar las técnicas apropiadas, se utilizó plantillas para describir el alcance de su función en el sistema a desarrollar. A continuación se muestran a manera de ejemplo algunas plantillas llenas con la información solicitada.

<b>Líder de la actividad de Elicitación</b>	
<b>Funciones</b>	<b>Interaccionar con el Cliente, Obtener información del cliente, atender sus inquietudes, interpretar la información proporcionada por los usuarios. Seleccionar las herramientas y técnicas adecuadas para la elicitación</b>
<b>Nombre</b>	<b>Rosa Maria Jurupe García</b>
<b>Cargo</b>	<b>Analista de Sistemas</b>
<b>Teléfono</b>	<b>4244168</b>
<b>Mail</b>	<b><a href="mailto:rjurupe@yahoo.com">rjurupe@yahoo.com</a></b>
<b>Nombre</b>	<b>Paola Pacheco Ramírez</b>
<b>Cargo</b>	<b>Analista de Sistemas</b>
<b>Teléfono</b>	<b>4231659</b>
<b>Mail</b>	<b><a href="mailto:pacheco@yahoo.com">pacheco@yahoo.com</a></b>

#### **2 Obtener Información sobre el dominio**

Para este caso se utilizó una entrevista con el cliente, coordinado fecha y hora. Se trata que sea una conversación abierta y general, tratando de entender el dominio del problema, Realizando preguntas abiertas, dejando al cliente que "cuente" lo que quiere o sea su problema.

Establecemos un plan de comunicación, intercambiando teléfonos, celulares, direcciones de e-mail correspondientes, horarios de contacto. Establecemos un cronograma para las entrevistas, con respecto a esto punto recomiendo realizarlas en un principio por lo menos una vez por semana. Esto depende del cliente con el que se este trabajando, en nuestro caso lo realizamos de esta forma y nos fue de gran utilidad ya que si transcurre más de una semana entre una entrevista y otra se pierde el hilo del proyecto.

Participantes: AN, GEPRO, ARQ, SQA, CLIENTE

Duración: 2:30 hs. Aprox.

<b>Técnica</b>	Preguntas generales, Preguntas abiertas.
<b>Herramientas</b>	Acta de reunión. Cuestionario.

### **3 Preparar y realizar las reuniones de elicitación**

Nuevamente se realiza una entrevista con el cliente, habiendo coordinado preliminarmente fecha y hora. Continuamos con una charla abierta, grabando (formato audio) la entrevista, para luego ser utilizada en la fase de análisis. Se hablan temas que van desde el funcionamiento del negocio, hasta requerimientos propios del sistema. Es difícil distinguir en el momento mismo de realizar la entrevista que es una cosa y cual es otra; para reducir el riesgo de malos entendidos manejamos la grabación como herramienta de registro, ya que de esta manera no se pierde parte alguna de la conversación y no se corta la fluidez de la comunicación a través de la escritura de todos los puntos tratados.

También es bueno averiguar si se disponen de procesos definidos, con esto me refiero a que los mismos se encuentren explícitamente documentados. Siempre se poseen procesos para hacer las cosas (cuando preguntamos ¿cómo haces tal cosa?, explicaran lo que hacen, en definitiva es el proceso). Esta información es de suma importancia para comprender como funciona la empresa.

**Participantes:** AN, CLIENTE

**Duración:** 2:00 hs. Aprox.

<b>Técnica</b>	Conversaciones abiertas. AN actúa pasivamente. Se basa en un cuestionario con preguntas abiertas.
<b>Herramientas</b>	Pizarrón, Grabador- Formato audio digital.

#### **4 Identificar los Objetivos del Sistema**

En nuestro caso práctico se han seleccionado e identificado cinco objetivos del sistema a manera de ejemplo, registrándolos en las plantillas de objetivo propuesta en la sección **¡Error! No se encuentra el origen de la referencia.** (Pagina **¡Error! Marcador no definido.**)

A continuación se listan los objetivos del Sistema:

<b>Identificador</b>	<b>Nombre</b>
<b>OBJ-001</b>	<b>Gestionar órdenes de compra de la distribuidora.</b>
<b>OBJ-002</b>	<b>Mantener actualizado el stock en los almacenes.</b>
<b>OBJ-003</b>	<b>Mantener actualizado el costo promedio de los materiales por cada movimiento de ingreso o egreso en los almacenes.</b>
<b>OBJ-004</b>	<b>Gestionar la información de ingresos y egresos de materiales al almacén.</b>
<b>OBJ-005</b>	<b>Gestionar Documentos de los proveedores</b>

**Tabla 4.11: Tabla de Listado de los objetivos del Sistema**

A Continuación se registran los objetivos identificados en las plantillas de Objetivos:

**OBJ-001: EL SISTEMA DEBE GESTIONAR LAS ÓRDENES DE COMPRA DE LA DISTRIBUIDORA**

<b>OBJ-001</b>	<b>Gestionar órdenes de compra de la distribuidora.</b>
----------------	---

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

<b>Versión</b>	000 (29/08/2007)
<b>Autores</b>	Paola Pacheco (Consultora Y) Rosa Jurupe (Consultora Y)
<b>Descripción</b>	<p>Gestionar ordenes de compra por parte de las distribuidoras , por razones diferentes:</p> <ul style="list-style-type: none"> <li>– Por compras periódicas de acuerdo al stock.</li> <li>– No se contempló en la gestión de stock o en los planes de ventas.</li> <li>– Por cantidades pedidas fuera de lo común para algún o algunos clientes.</li> </ul> <p>En él se puede colocar todos los materiales que la distribuidora solicita.</p>
<b>Importancia</b>	01
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	<p>Los usuario desean en el registro de ordenes de compra una rápida búsqueda de los materiales, dando la opción de indicar el material en unidades de medida base o unidades de medida de venta.</p> <p>El plan de venta es un proyectado de lo que debería de vender la distribuidora.</p> <p>Esto es elaborado por el proveedor.</p>

**OBJ-002: EL SISTEMA DEBE MANTENER ACTUALIZADO EL STOCK DE MATERIALES EN LOS ALMACENES**

<b>OBJ-002</b>	<b>Mantener actualizado el stock en los almacenes.</b>
<b>Versión</b>	000 (29/08/2007)
<b>Autores</b>	Paola Pacheco (Consultora Y) Rosa Jurupe (Consultora Y)
<b>Descripción</b>	En la ejecución de cada movimiento (ingresos por compras, traslado entre almacenes, ingresos o egresos por otros motivos, etc.) en los almacenes se debe mantener actualizada

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

	inmediatamente la información del stock de cada material por almacén en el sistema.
<b>Importancia</b>	2
<b>Estabilidad</b>	Alta

**OBJ-003: EL SISTEMA DEBE POSEER INFORMACIÓN ACTUALIZADA DEL COSTO PROMEDIO DE LOS MATERIALES POR CADA MOVIMIENTO EN LOS ALMACENES.**

<b>OBJ-003</b>	Mantener información actualizada del costo promedio de los materiales por cada movimiento de ingreso o egreso en los almacenes.
<b>Versión</b>	000 (29/08/2007)
<b>Autores</b>	Paola Pacheco (Consultora Y)
<b>Descripción</b>	En la ejecución de cada movimiento (ingresos por compras, traslado entre almacenes, ingresos o egresos por otros motivos, etc.) en los almacenes se debe de obtener el nuevo costo promedio del material en el almacén y ser registrado en el sistema.
<b>Importancia</b>	3
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Las formulas de obtener costo promedio por cada movimiento serán dadas por la gente de sistemas.

**OBJ-004: EL SISTEMA DEBE GESTIONAR LA INFORMACIÓN DE INGRESOS Y EGRESOS DE MATERIALES AL ALMACÉN**

<b>OBJ-004</b>	Gestionar la información de ingresos y egresos de materiales al almacén.
<b>Versión</b>	000 (29/08/2007)

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

<b>Autores</b>	Paola Pacheco (Consultora Y)
<b>Descripción</b>	Gestionar en el sistema los partes (documento de entrada o salida de movimientos de los materiales en los almacenes de la distribuidora. En cada parte generado es indispensable indicar el tipo de movimiento que se realizó.
<b>Importancia</b>	04
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Se obtuvo formato de partes de movimientos anteriores.

**OBJ-005: EL SISTEMA DEBE GESTIONAR INFORMACIÓN DE TODOS LOS DOCUMENTOS RELACIONADOS A LA COMPRA A PROVEEDORES**

<b>OBJ-005</b>	<b>Gestionar información de documentos de compra de los proveedores</b>
<b>Versión</b>	000 (29/08/2007)
<b>Autores</b>	Paola Pacheco (Consultora Y)
<b>Descripción</b>	Se debe mantener registrado en el sistema datos de los documentos de los proveedores que han sido recepcionados por la distribuidora.
<b>Importancia</b>	5
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	En las entrevistas se pudo conseguir documentos como factura por compras y factura por transporte.

## **5 Clasificar los requerimientos**

A continuación clasificamos los algunos de los requerimientos elicitados a manera de ejemplo, en funcionales, almacenamiento y restricciones, listamos cada uno de los requerimientos de acuerdo al grupo de clasificación y por ultimo los registramos en la plantilla correspondiente a tipo de requerimiento.



**Lista de Requerimientos de Almacenamiento de Información del Modulo Almacén**

<b>identificador</b>	<b>Nombre</b>
<b>IRQ-001</b>	<b>Información Documentos de Compra proveedor</b>
<b>IRQ-002</b>	<b>Información de partes de movimientos</b>
<b>IRQ-003</b>	<b>Información el material en el almacén</b>
<b>IRQ-004</b>	<b>Información de almacén</b>
<b>IRQ-005</b>	<b>Información del Proveedor</b>
<b>IRQ-006</b>	<b>Información de personal-Almacén</b>
<b>IRQ-007</b>	<b>Información Centro Distribución</b>
<b>IRQ-008</b>	<b>Información Maestro Materiales</b>
<b>IRQ-009</b>	<b>Información de Sedes</b>
<b>IRQ-010</b>	<b>Información de Distribuidoras</b>

**Lista de restricciones de los requerimientos de almacenamiento de Información del Modulo de almacén**

<b>Identificador</b>	<b>Nombre</b>
<b>CRQ-001</b>	<b>Parte de movimientos de materiales es por un almacén</b>
<b>CRQ-002</b>	<b>El ingreso de factura de compras se hace una sola vez.</b>

**IRQ-001: PLANTILLA DE ALMACENAMIENTO DE INFORMACIÓN DE DOCUMENTOS DE COMPRA DEL PROVEEDOR**

<b>IRQ-001</b>	<b>Información Documentos de Compra proveedor</b>
<b>Versión</b>	00 (29/08/2007)
<b>Autores</b>	Paola Pacheco (Consultora Y)
<b>Descripción</b>	<p>El sistema debe tener la información necesaria sobre la factura de compras (documento entregado por el proveedor). Los datos de la factura de compras son las siguientes:</p> <p>Cabecera, Número del documento del proveedor, Número preimpreso</p> <p>Fecha de Emisión, Fecha de liquidación , Fecha de vencimiento</p> <p>Código de la orden de compra, Código del tipo del documento, Código del Proveedor, Importe Total bruto, Importe Total otros gastos, Importe Total de los impuestos, Importe neto del pedido, Código de moneda</p> <p>Código de la distribuidora, Código de la sede, Código de la oficina de ventas.</p> <p>Detalle : Código de Material, Código de unidad de medida ,Cantidad, Importe unitario , Importe total de descuento, Importe total</p> <p>Así mismo a la factura se le debe considerar estados como: Pendiente, ingresado, anulado.</p>
<b>Objetivos asociados</b>	OBJ-005 : Gestionar documentos de los proveedores
<b>Requerimiento asociados</b>	<p>CU-001 : Registrar facturas por compras</p> <p>CU-002 : Registrar ingreso por compras</p>

**CRQ-001 PLANTILLA DE RESTRICCIÓN DE ALMACENAMIENTO DE INFORMACIÓN DE LOS PARTES DE MOVIMIENTO POR ALMACÉN**

<b>CRQ-001</b>	<b>Parte de movimientos de materiales es por un almacén.</b>
----------------	--

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

<b>Versión</b>	00 (29/08/2007)
<b>Autores</b>	Paola Pacheco (Consultora Y)
<b>Descripción</b>	<p>Los movimientos de ingresos y egresos de materiales en los almacenes originarán partes de movimientos de materiales. Pero cada parte debe estar asociado a un almacén.</p> <p>Ejemplo:</p> <p>En el caso de ingreso por factura de compras vendrán materiales que se pueden ir a diferentes almacenes dependiendo de la condición de almacenaje del material. Entonces el ingreso de factura de compras originará partes de movimientos por cada almacén donde se hizo el ingreso de materiales.</p>
<b>Objetivos asociados</b>	OBJ-004: Administrar información de los ingresos y egresos de materiales al almacén.
<b>Requerimiento asociados</b>	<p>CU-002 : Registrar ingreso por compras</p> <p>CU-003: Registrar traslado de materiales entre almacenes</p> <p>CU-004: Registrar ingresos por otros motivos</p> <p>CU-005: Ajustar Inventario</p> <p>CU-006: Cambiar estado del Material</p> <p>CU-007: Registrar partes de movimientos de materiales</p>

**Lista de Requerimientos Funcionales del Modulo Almacén**

Identificador	Nombre
<b>CU-001</b>	<b>Registrar factura de compra</b>
<b>CU-002</b>	<b>Registrar ingreso por compras</b>
<b>CU-003</b>	<b>Traslado entre almacenes</b>
<b>CU-004</b>	<b>Registrar ingresos por otros motivos</b>
<b>CU-005</b>	<b>Ajustar Inventario</b>
<b>CU-006</b>	<b>Cambiar estado del Material(de calidad a disponible, de disponible a calidad)</b>
<b>CU-007</b>	<b>Registrar partes de movimientos de materiales</b>

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

<b>CU-008</b>	<b>Valorizar Costo Promedio</b>
<b>CU-009</b>	<b>Actualizar stock de material</b>
<b>CU-010</b>	<b>Reversión de movimientos de materiales</b>
<b>CU-011</b>	<b>Consultar stock</b>
<b>CU-012</b>	<b>Generar Reporte de Stock Valorizado</b>
<b>CU-013</b>	<b>Consulta de Partes</b>

**REGISTRAR INGRESO DE DOCUMENTOS POR COMPRAS**

<b>FRQ-002</b>	Registrar ingreso por compras
<b>Versión</b>	0
<b>Autores</b>	Paola Pacheco Rosa Jurupe
<b>Fuente</b>	Área Almacén de Distribuidora Gavialdi Administrador José Perez Gerente: Mario Sanchez
<b>Caso Uso Asociado</b>	Registrar ingreso por compras

CU-002	Registrar ingreso por compras
DESCRIPCION	Ingreso que solicitará el número de la Factura que proviene del proveedor para el ingreso de materiales por compras.
ACTOR(ES)	Almacenero, Asistente administrativo
PRE – CONDICIONES	<p>El usuario ha ingresado al sistema y cuenta con la autorización para realizar esta operación.</p> <p>Se ha registrado en el sistema la Factura de la compra.</p>
POST – CONDICIONES	Se ha registrado el ingreso de los materiales en el almacén.
ESCENARIO PRIMARIO	Registrar ingreso por compras con la factura recibida.
DESCRIPCIÓN DEL FLUJO	
Actor	Sistema
1. El usuario ingresa a la opción “Ingreso por Compras”.	2. El sistema muestra una pantalla donde el usuario puede ingresar el número de Factura generado por el proveedor.
3. El usuario ingresa el número de la factura y presiona el botón Aceptar.	4. Se ejecuta el caso de uso “Consultar Facturas de Compras” con el número de Factura ingresado. 5. El sistema muestra la Factura como no editable.
6. El usuario acepta el ingreso total y presiona el botón “Registrar”.	7. El sistema ejecuta el caso de uso “Convertir Unidad de Medida”, pasará los siguientes parámetros: código de material, cantidad y unidad de medida de la factura, y se obtiene la cantidad y la unidad de medida base. 8. El sistema calcula el costo promedio

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

		<p>de los materiales con la cantidad en unidad de medida base, caso de uso “Valorizar Stock”</p> <p>9. “Se ejecuta el caso de uso “Registrar Parte de movimiento de materiales”, pasar los parámetros: Cantidades en unidad de medida base, costo promedio, número de factura, tipo de movimiento y materiales.</p> <p>10. El sistema muestra la opción de imprimir el parte.</p>
<b>EXCEPCIONES</b>		La factura no existe
		1. El sistema muestra el mensaje de error: “La factura no existe”
<b>ESCENARIO SECUNDARIO</b>		Realizar la Reversión del Parte de compra
<b>DESCRIPCIÓN DEL FLUJO</b>		
<b>Actor</b>		<b>Sistema</b>
1. El usuario ingresa a la opción “Reversión de Parte de Compra”.		2. El sistema ejecuta el caso de uso “Consulta de Partes”.
3. El usuario ingresa las observaciones de la reversión y presiona el botón Aceptar		<p>4. El sistema ejecuta el caso de uso “Registrar Parte de movimiento de materiales” indicando el tipo de movimiento (salida de materiales).</p> <p>5. El sistema calcula el costo promedio con la cantidad en unidad de medida base, caso de uso “Valorizar Stock”.</p>

**OPERACIÓN DE TRASLADO DE MATERIALES ENTRE ALMACENES**

<b>FRQ-003</b>	Traslado entre almacenes
----------------	--------------------------

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

<b>Versión</b>	0
<b>Autores</b>	Paola Pacheco Rosa Jurupe
<b>Fuente</b>	Área Almacén de Distribuidora Gavialdi Administrador José Perez Gerente: Mario Sanchez
<b>Caso Uso Asociado</b>	Traslado de Materiales entre almacenes

CU-003	Traslado entre almacenes	
DESCRIPCION	Permitirá realizar el traslado de materiales entre dos almacenes de un mismo centro de distribución.	
ACTOR(ES)	Almacenero, Asistente administrativo	
PRECONDICIONES	–	El usuario ha ingresado al sistema y cuenta con la autorización para realizar esta operación. Esta transferencia es entre almacenes de un mismo centro de distribución.
POSTCONDICIONES	–	Se ha registrado el ingreso de los materiales en el almacén.
ESCENARIO PRIMARIO	Trasladar entre almacenes	
DESCRIPCIÓN DEL FLUJO		
Actor		Sistema
1. El usuario ingresa a la opción “Traslado entre almacenes”		2. El sistema muestra una pantalla de selección: <ul style="list-style-type: none"><li>• Lista de distribuidoras (sólo para usuario)</li></ul>

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

	<ul style="list-style-type: none"> <li>• Lista de sedes</li> <li>• Almacén origen</li> <li>• Almacén destino</li> <li>• El código de material, le permitirá realizar una búsqueda del material (mayor detalle ver caso de uso "Buscar Material").</li> </ul>
<p>3. El usuario recibe una página e ingresa:</p> <ul style="list-style-type: none"> <li>• La distribuidora</li> <li>• La Sede</li> <li>• Centro de distribución (se obtiene automáticamente)</li> <li>• Código de almacén origen</li> <li>• Código de almacén destino</li> </ul> <p>Para cada material:</p> <ul style="list-style-type: none"> <li>• Código de material</li> <li>• Cantidad</li> <li>• Unidad de medida</li> </ul> <p>4. El usuario presionará el botón Aceptar.</p>	<p>5. El sistema consultará los datos de stock del material seleccionado para ello ejecutará el caso de uso "Consultar Material" con Parámetros: Código de material, Centro de distribución, Código de almacén</p> <p>6. El sistema mostrará en una pantalla la siguiente información:</p> <ul style="list-style-type: none"> <li>• Código de la Distribuidora</li> <li>• Código la Sede</li> <li>• Código de centro de distribución y descripción</li> <li>• Código del almacén y descripción</li> </ul> <p>Para cada material</p> <ul style="list-style-type: none"> <li>• Código de material y descripción</li> <li>• Unidad de medida base</li> <li>• Stock disponible</li> <li>• Stock comprometido</li> </ul>
<p>7. El usuario presiona el botón "Registrar".</p>	<ul style="list-style-type: none"> <li>• El sistema calcula el costo promedio de los materiales del almacén de origen, caso de uso "Valorizar Stock"</li> </ul>



**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

		<div>8. Se ejecuta el caso de uso “Registrar Parte de movimiento de materiales” indicando el tipo de movimiento(salida por traslado).</div> <div>9. El sistema muestra la opción de imprimir el Parte.</div>
<div>ESCENARIO SECUNDARIO</div>	Realizar la Reversión del traslado entre almacenes	
DESCRIPCIÓN DEL FLUJO		
Actor	Sistema	
<div>1. El usuario ingresa a la opción “Reversión del Traslado entre almacenes”.</div>	<div>2. El sistema ejecuta el caso de uso “Consulta de Partes”.</div>	
<div>3. El usuario ingresa las observaciones de la reversión y presiona el botón Aceptar</div>	<div>4. El sistema ejecuta el caso de uso “Registrar Parte de movimiento de materiales” indicando como movimiento de ingreso.</div> <div>5. El sistema calcula el costo promedio de los materiales del almacén, caso de uso “Valorizar Stock”</div>	

**OPERACIÓN DE REGISTRO DE PARTES DE MOVIMIENTO DE MATERIALES**

<b>FRQ-007</b>	Registrar Parte de movimiento de materiales
<b>Versión</b>	0
<b>Autores</b>	Paola Pacheco Rosa Jurupe
<b>Fuente</b>	Área Almacén de Distribuidora Gavialdi Administrador José Perez

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

	Gerente: Mario Sanchez
<b>Caso Uso Asociado</b>	Registrar Parte de movimiento de materiales

CU-007	Registrar Parte de movimiento de materiales	
DESCRIPCION	Registra los movimientos de materiales del almacén que son originados por ingresos de compras, salidas por ventas, traslado y devolución.	
ACTOR(ES)	Sistema ó Aplicación	
PRE CONDICIONES	–	El usuario ha ingresado al sistema y cuenta con la autorización para realizar esta operación.
POST CONDICIONES	–	Se ha creado el parte y registrado el movimiento de materiales del almacén.
ESCENARIO PRIMARIO	Registrar Parte de movimiento de materiales.	
DESCRIPCIÓN DEL FLUJO		
Actor		Sistema
1. El sistema invoca al proceso de registro de Parte de movimiento de materiales.		2. El sistema recibe como parámetros: <ul style="list-style-type: none"><li>• Tipo de movimiento<ul style="list-style-type: none"><li>○ Compras</li><li>○ Ventas</li><li>○ Ingreso por Traslado</li><li>○ Salida por Traslado</li><li>○ Salida por Devolución a Proveedor</li><li>○ Entrada por Devolución del cliente</li><li>○ Ajuste de material faltante</li></ul></li></ul>

	<ul style="list-style-type: none"> <li>○ Ajuste de material sobrante</li> <li>• Tipo de documento asociado <ul style="list-style-type: none"> <li>○ Factura</li> <li>○ Boleta</li> <li>○ Guía de Remisión</li> </ul> </li> <li>• Lista de materiales y cantidades (si el tipo es Devolución)</li> </ul> <p>3. El sistema obtiene los materiales asociados al documento de referencia(si no es devolución) y se realizan las siguientes validaciones:</p> <ul style="list-style-type: none"> <li>• Si el tipo de movimiento es “Venta” o “Salida por Traslado”, se validará que las unidades a mover sean menores o iguales al stock comprometido.</li> <li>• Si el tipo de movimiento es “Devolución” o “Salida por Traslado” se validará que las unidades a mover sean menores o iguales al stock disponible.</li> </ul> <p>4. El sistema crea el Parte y registra el movimiento de materiales.</p> <ul style="list-style-type: none"> <li>• Si el tipo de movimiento es de “Compras” , “Ingreso por Traslado”, “Entrada por Devolución de Cliente” o “Ajuste por sobrante” se incrementa el stock de libre disponibilidad.</li> <li>• Si el tipo de movimiento es “Venta”, “Salida por Traslado” se disminuirá el stock comprometido.</li> </ul>
--	--

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

	<ul style="list-style-type: none"> <li>• Si el tipo de movimiento es “Salida por Traslado” se disminuye el stock disponible y se aumenta el stock en tránsito en el almacén de destino (diferente centro de distribución).</li> <li>• Si el tipo de movimiento es “Ajuste por faltante” se disminuye el stock disponible.</li> <li>• Si el tipo de movimiento es “Salida por devolución a Proveedor” se disminuye el stock disponible.</li> <li>• Se registra la fecha de creación, tipo de parte, tipo de movimiento, documento asociado, el almacén respectivo, responsable del almacén, usuario de creación del parte</li> </ul> <p>5. Se ejecuta el caso de uso “Registrar operación para contabilidad”.</p>
<b>EXCEPCIONES</b>	El stock comprometido no es suficiente
	<p>1. En el punto 2 del flujo primario en caso que no pase la validación se devolverá un mensaje de error “El stock comprometido no es suficiente para registrar la salida de X unidades del material Y”.</p>
<b>EXCEPCIONES</b>	El stock disponible no es suficiente
	<p>1. En el punto 2 del flujo primario en caso que no pase la validación se devolverá un mensaje de error “El</p>

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

	stock disponible no es suficiente para registrar la salida de X unidades del material Y”.
--	---

**OPERACIÓN DE VALORIZACION DE COSTO PROMEDIO DE LOS MATERIALES**

<b>FRQ-008</b>	Valorizar Costo Promedio
<b>Versión</b>	0
<b>Autores</b>	Paola Pacheco Rosa Jurupe
<b>Fuente</b>	Área Almacén de Distribuidora Gavialdi Administrador José Perez Gerente: Mario Sanchez
<b>Caso Uso Asociado</b>	Valorizar Costo Promedio

<b>CU-008</b>	<b>Valorizar Costo Promedio</b>
<b>DESCRIPCION</b>	Calcular del costo promedio por operación
<b>ACTOR(ES)</b>	Sistema ó Aplicación
<b>PRE CONDICIONES</b>	El usuario ha ingresado al sistema y cuenta con la autorización para realizar esta operación.
<b>POST CONDICIONES</b>	Ninguno
<b>ESCENARIO PRIMARIO</b>	Calcular costo promedio por operación de ingreso material por compras o traslado( para el almacén de destino)

DESCRIPCIÓN DEL FLUJO	
Actor	Sistema
	<ol style="list-style-type: none"> <li>1. Recibe como parámetros: <ul style="list-style-type: none"> <li>• Código de material</li> <li>• Cantidad ingresada en unidades (<math>Cant_2</math>)</li> <li>• Costo del material (<math>Costo_2</math>)</li> </ul> </li> <li>2. El sistema consulta el último costo promedio del material (<math>CostoPr om_1</math>)</li> <li>3. El sistema consulta el stock total del material (<math>Cant_1</math>).</li> <li>4. El sistema realiza la siguiente operación: <math display="block">CostoPr om \equiv \left( \frac{Cant_1 * CostoPr om_1 + Cant_2 * Costo_2}{Cant_1 + Cant_2} \right)</math> </li> <li>5. Se devuelve el costo promedio obtenido.</li> </ol>
<b>ESCENARIO PRIMARIO</b>	Calcular costo promedio por registro de N/C por devolución de compra o N/C por valor
DESCRIPCIÓN DEL FLUJO	
Actor	Sistema
	<ol style="list-style-type: none"> <li>1. Recibe como parámetros: <ul style="list-style-type: none"> <li>• Código de material</li> <li>• Cantidad devuelta en unidades (<math>Cant_2</math>)</li> <li>• Costo del material (<math>Costo_2</math>)</li> </ul> </li> <li>2. El sistema consulta el último costo promedio del material (<math>CostoPr om_1</math>)</li> </ol>

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

	<div>3. El sistema consulta el stock total del material (<math>Cant_1</math>).</div> <div>4. El sistema realiza la siguiente operación: <math display="block">CostoPr om \equiv \left( \frac{Cant_1 * CostoPr om_1 - Cant_2 * Costo_2}{Cant_1 - Cant_2} \right)</math></div> <div>5. Se devuelve el costo promedio obtenido.</div>
ESCENARIO PRIMARIO	Calcular costo promedio por registro de N/D por valor
DESCRIPCIÓN DEL FLUJO	
Actor	Sistema
	<div>1. Recibe como parámetros:<ul style="list-style-type: none"><li>• Código de material</li><li>• Cantidad recibida en unidades (<math>Cant_2</math>)</li><li>• Costo del material (<math>Costo_2</math>)</li></ul></div> <div>2. El sistema consulta el último costo promedio del material (<math>CostoPr om_1</math>)</div> <div>3. El sistema consulta el stock total del material (<math>Cant_1</math>).</div> <div>4. El sistema realiza la siguiente operación: <math display="block">CostoPr om \equiv \left( \frac{Cant_1 * CostoPr om_1 + Cant_2 * Costo_2}{Cant_1 + Cant_2} \right)</math></div> <div>5. Se devuelve el costo promedio obtenido.</div>

OPERACIÓN DE CONSULTA DE STOCK DE LOS MATERIALES

<b>FRQ-009</b>	Consulta de stock del material
<b>Versión</b>	0
<b>Autores</b>	Paola Pacheco Rosa Jurupe
<b>Fuente</b>	Área Almacén de Distribuidora Gavialdi Administrador José Perez Gerente: Mario Sanchez
<b>Caso Uso Asociado</b>	Consulta de stock del material

<b>CU-009</b>	<b>Consulta de stock del material</b>
<b>DESCRIPCIÓN</b>	Realiza una consulta de stock
<b>ACTOR(ES)</b>	Almacenero, Asistente administrativo
<b>PRE CONDICIONES</b>	El usuario ha ingresado al sistema y cuenta con la autorización para realizar esta operación.
<b>POST CONDICIONES</b>	Ninguna
<b>ESCENARIO PRIMARIO</b>	Consultar stock
<b>DESCRIPCIÓN DEL FLUJO</b>	



**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

<b>Actor</b>		<b>Sistema</b>
1. El usuario selecciona la opción “Gestión de Almacenes” y “Consultar stock”.		2. El sistema muestra un formulario con los siguientes filtros de búsqueda: <ul style="list-style-type: none"> <li>• Distribuidoras</li> <li>• Sede</li> <li>• Centro de Distribución</li> <li>• Almacén</li> <li>• Familia</li> <li>• Categoría</li> <li>• Código de material</li> <li>• Nombre de material</li> </ul>
3. El usuario ingresa los filtros de búsqueda y presiona el botón “Buscar”		4. El sistema muestra los resultados de la búsqueda en una lista que contiene las siguientes columnas: <ul style="list-style-type: none"> <li>• Distribuidora</li> <li>• Sede</li> <li>• Centro de distribución</li> <li>• Almacén</li> <li>• Código de material</li> <li>• Unidad de medida base</li> <li>• Nombre de material</li> <li>• Stock disponible</li> <li>• Stock comprometido</li> <li>• Stock en tránsito.</li> <li>• Stock en control de calidad</li> <li>• Stock Total (stock físico en el almacén)</li> </ul>
<b>EXCEPCIONES</b>		No se encontraron registros
		1. El sistema muestra el mensaje “No se encontraron registros”.

**Matriz de Objetivos vs. Requerimientos de almacenamiento de Información y**

	OBJ-001	OBJ-002	OBJ-003	OBJ-004	OBJ-005
IRQ-001	X	X	X	X	X
IRQ-002			X	X	
IRQ-003		X	X	X	
IRQ-004		X	X	X	
IRQ-005	X				X
IRQ-006	X	X	X	X	X
IRQ-007	X	X	X	X	
IRQ-008	X				X
IRQ-009	X	X	X	X	
IRQ-010	X	X	X	X	X
CU-001					X
CU-002		X	X	X	
CU-003		X	X	X	
CU-004		X	X	X	
CU-005		X	X	X	
CU-006		X	X	X	
CU-007				X	
CU-008			X		
CU-009		X		X	
CU-010		X	X	X	
CU-011		X			
CU-012			X	X	
CU-013				X	

**Funcionales**

**6 Evaluar y racionalizar**

Estas tareas las realizamos juntamente cuando registramos en las plantillas detectando requerimientos inconsistentes que provienen de la obtención de los requerimientos por las técnicas de entrevistas.

**7 Dar Prioridad**

Esta tarea también se realizó en paralelo con el registro de los requerimientos en las plantillas de tal manera que las plantillas muestran la prioridad en uno de sus patrones.

**8 Integrar y Validar**

En esta tarea se revisaron los requerimientos registrados juntamente con el Cliente participando este en la documentación de sus requerimientos.

**9 Documentar etapa**

Se realizó el primer bosquejo de nuestro documento de requerimientos agrupando todas las plantillas revisadas por los stakeholders y agrupándolos para que entren a la actividad de análisis y negociación.

**4.8.2 Utilizando la metodología propuesta para la actividad el Análisis de requerimientos**

**1. Analizar los requerimientos de almacenamiento de Información**

De acuerdo a esta tarea realizamos el diagrama de clases de los requerimientos de almacenamiento de información utilizando la notación de UML

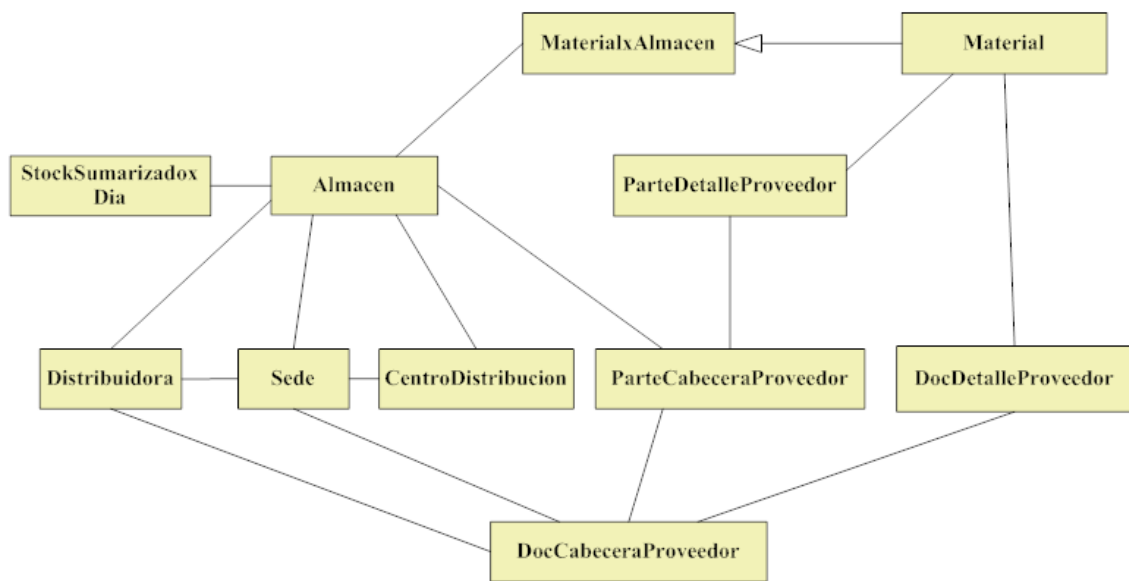


Figura 4.5: Diagrama de Clases del Modulo de Almacén

#### LISTA DE CLASES IDENTIFICADAS A PARTIR DE LOS REQUERIMNENTOS DE ALMACENAMIENTO DE INFORMACIÓN

Identificador	Nombre
<b>MaterialxAlmacen</b>	Listado de materiales por almacén
<b>Material</b>	Maestro de materiales
<b>Almacén</b>	Listado de almacenes de distribuidoras
<b>ParteCabProveedor</b>	Cabecera de parte de movimiento de materiales en almacén
<b>ParteDetalleProveedor</b>	Detalle de parte de movimiento de materiales en almacén
<b>DocCabeceraProveedor</b>	Cabecera de documento de compra de proveedor
<b>DocDetalleProveedor</b>	Detalle de documento de compra de proveedor
<b>Sede</b>	Información de Sede
<b>Centro</b>	Información de Centro

### PLANTILLA DE LA CLASE MATERIALXALMACEN

Clase	<i>MaterialxAlmacen</i>
<b>Versión</b>	IRQ-003 : Información de material en el almacén
<b>Autores</b>	Rosa Jurupe (Consultora Y)
<b>Requerimientos Asociados</b>	IRQ-003 : Información del material en el almacén CU-002 : Registrar ingreso por compras CU-003 : Registrar traslado de materiales entre almacenes CU-004 : Registrar ingresos por otros motivos CU-005 : Ajustar Inventario CU-006 : Cambiar estado del Material CU-007 : Registrar partes de movimientos de materiales CU-008 : Valorizar Costo Promedio CU-009 : Actualizar stock del material CU-010 : Reversión de Movimientos de Materiales CU-011 : Consultar stock
<b>Descripción</b>	Detalla la relación de materiales en el almacén de cada distribuidora.
<b>Superclases</b>	Material

### PLANTILLA DE LA CLASE MATERIAL

Clase	<i>Material</i>
<b>Versión</b>	00 (30/08/2007)
<b>Autores</b>	Rosa Jurupe (Consultora Y>)
<b>Requerimientos asociados</b>	IRQ-001: Información Documentos de Compra proveedor CU-004 : Registrar ingresos por otros motivos
<b>Descripción</b>	Tabla maestra de materiales, detalla la relación de materiales que el Proveedor puede vender a las distribuidoras.

**PLANTILLA DE LA CLASE PARTECABPROVEEDOR**

Clase	ParteCabProveedor
Versión	00 (30/08/2007)
Autores	Rosa Jurupe (Consultora Y)
Requerimientos asociados	IRQ-001: Información Documentos de Compra proveedor IRQ-003 : Información del material en el almacén CU-002 : Registrar ingreso por compras CU-003 : Registrar traslado de materiales entre almacenes CU-004 : Registrar ingresos por otros motivos CU-005 : Ajustar Inventario CU-006 : Cambiar estado del Material CU-007 : Registrar partes de movimientos de materiales CU-008 : Valorizar Costo Promedio CU-009 : Actualizar stock del material CU-010 : Reversión de Movimientos de Materiales
Descripción	Parte de movimientos de materiales en almacén.

**PLANTILLA DE ASOCIACION DE LA CLASE MATERIAL Y PARTEDETALLEPROVEEDOR**

Asociación	Material - ParteDetalleProveedor
Versión	00 (30/08/2007)
Autores	Rosa Jurupe (Consultora Y)
Descripción	En el detalle de parte de movimientos de materiales en el almacén se encontrarán materiales de la tabla maestra de materiales.

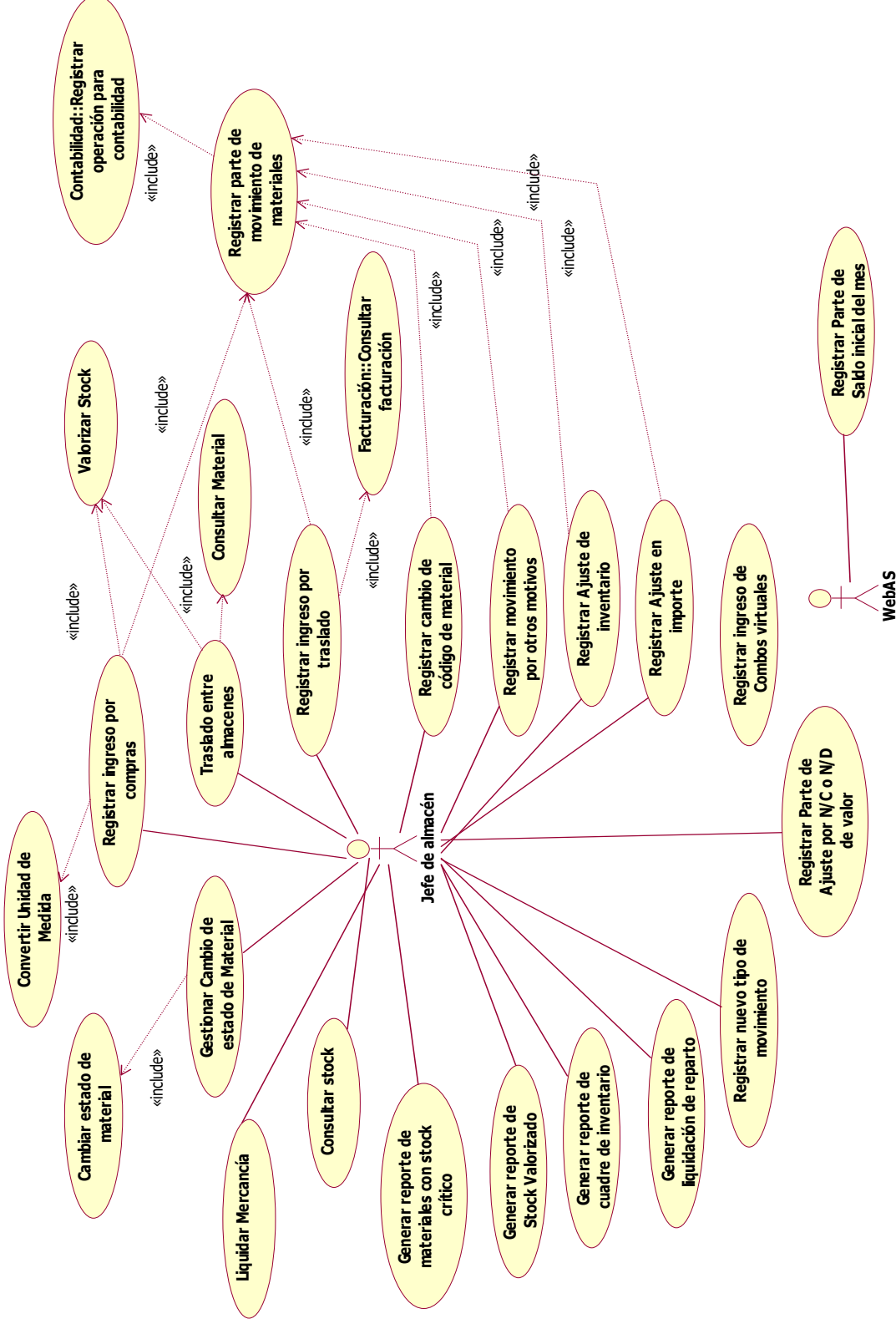
**PLANTILLA DE ASOCIACION DE LA CLASE PARTECABECERAPROVEEDOR Y DOCCABECERAPROVEEDOR**

Asociación	ParteCabeceraProveedor-DocCabeceraProveedor
Versión	00 (30/08/2007)
Autores	Rosa Jurupe (Consultora Y)
Descripción	La cabecera del parte de movimiento de materiales en materiales

	tendrá el número de documento de compra asociado.
--	---

## **2. Analizar los requerimientos funcionales del Modulo de Almacén**

En esta tarea hemos definido nuestro diagrama de Casos de Uso del Modulo de Almacén, las plantillas de Casos de uso ya se definieron anteriormente en los requerimientos funcionales.



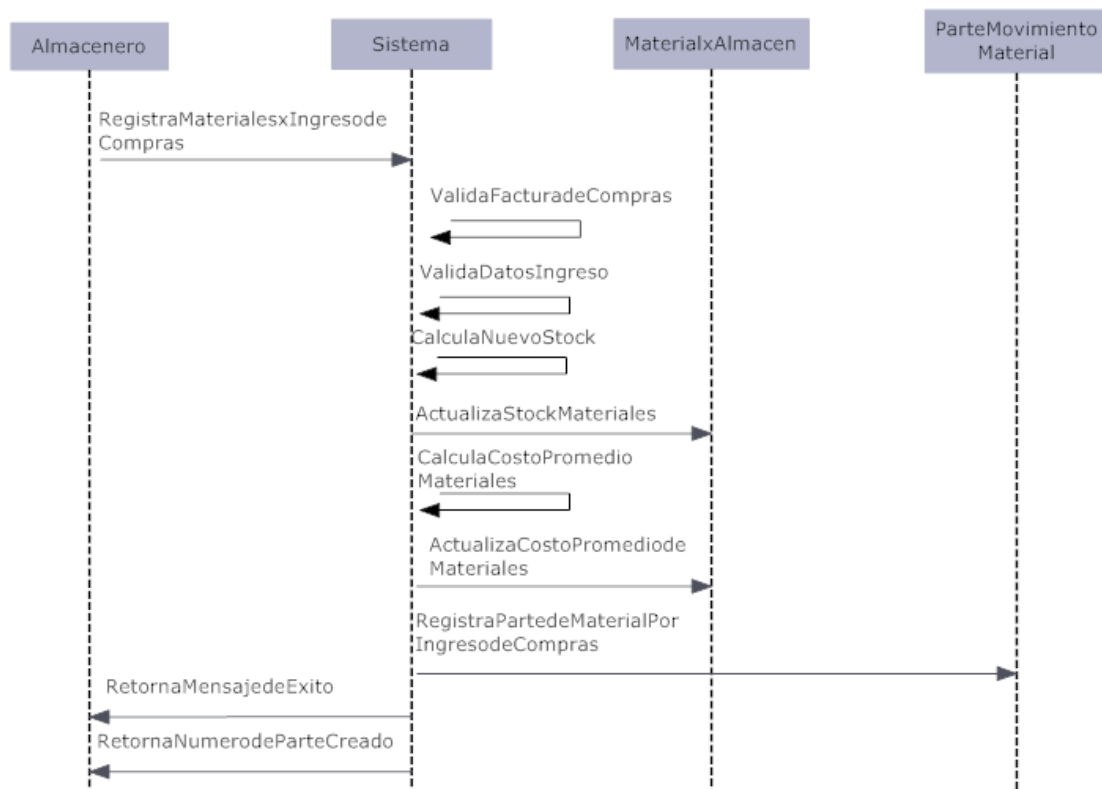


A continuación se utilizan las herramientas de diagrama de secuencia y de estados para entender la funcionalidad del:

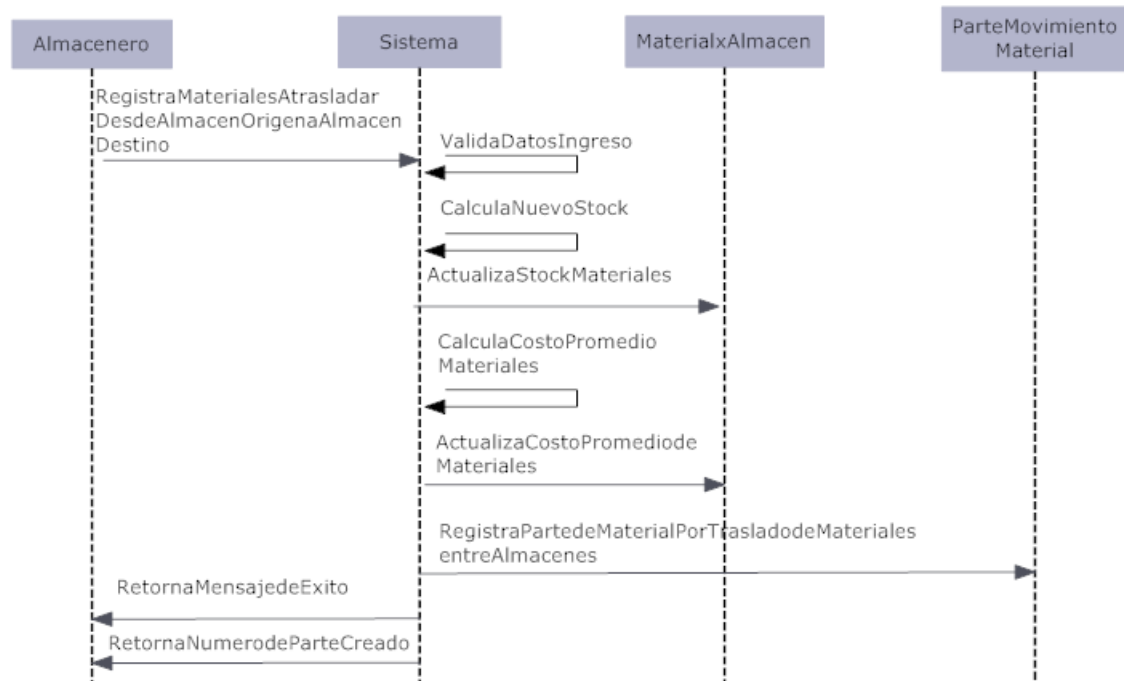
Registro de Documentos de Compras del Modulo de Almacén.

Traslado de Materiales entre almacenes

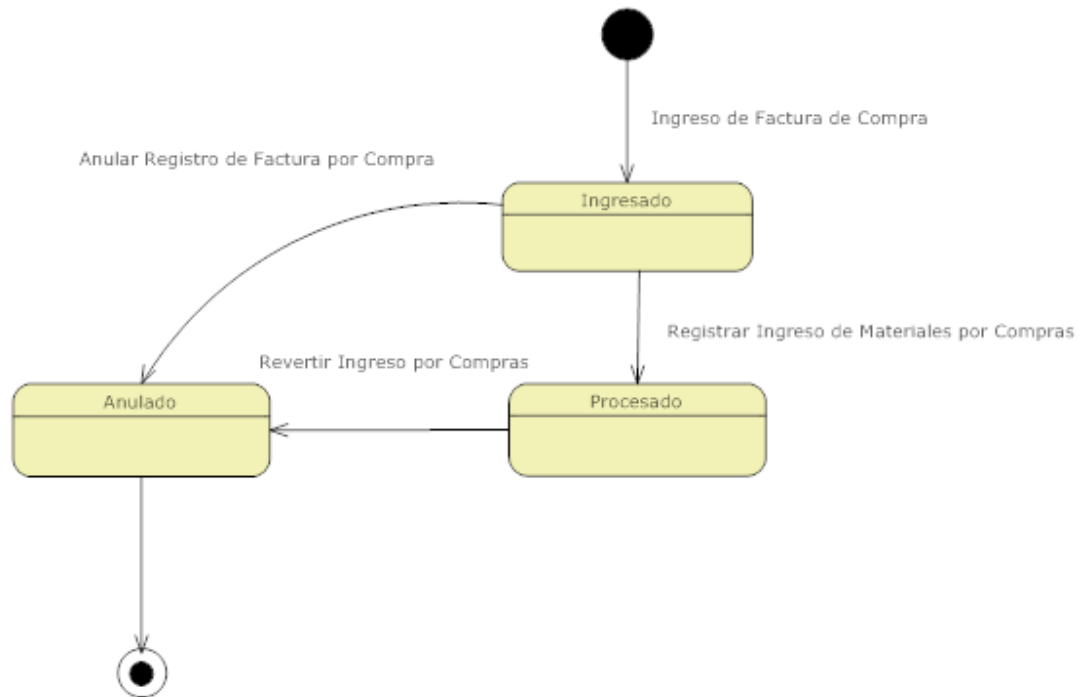
### Diagrama de Secuencia: Registrar Documento de Compra de los proveedores



### Diagrama de Secuencia: Traslado de materiales entre almacenes



**Diagrama de Estado: Registrar Documento de Compra de los Proveedores**



## **4.9 Herramientas automatizadas para la administración de Requerimientos**

Hoy en día, la Ingeniería de Software cuenta con una serie de herramientas automatizadas destinadas a diferentes propósitos. Dentro de las herramientas CASE que sirven de apoyo a los procesos de Ingeniería de Software, están las especializadas en la administración de requerimientos. Estas herramientas se concentran en capturar requerimientos, administrarlos y producir una especificación de requerimientos.

Las ventajas que nos proporcionan las herramientas automatizadas para IR son:

- Permiten un mayor control de proyectos complejos.
- Permiten reducir costos y retrasos en la liberación de un proyecto.
- Permiten una mayor comunicación en equipos de trabajo.
- Ayudan a determinar la complejidad del proyecto y esfuerzos necesarios.

En este capítulo veremos dos de las herramientas más utilizadas para este propósito: **RequisitePro y DOORS**.

### **4.9.1 RequisitePro**

RequisitePro<sup>(R)</sup> es la herramienta que ofrece Rational Software para tener un mayor control sobre los requerimientos planteados por el usuario y todos aquellos requerimientos técnicos o nuevos requerimientos de usuario que surjan durante el ciclo de vida del proyecto.

Con RequisitePro los requerimientos se encuentran documentados bajo un esquema organizado de documentos; estos esquemas, cumplen completamente con los estándares requeridos por IEEE, ISO, SEI, CMM y por el Rational Unified Process.

RequisitePro se integra con aplicaciones para la administración de cambios, herramientas de modelado de sistemas y con herramientas de pruebas. Esta

integración asegura que los diseñadores conocen los requerimientos del usuario, del sistema y del software en el momento de su desarrollo. RequisitePro permite el desarrollo en equipo, vía el check-in y check-out de los documentos involucrados en el proyecto. Con esta integración, se pueden conservar juntos todos los requerimientos y ser manipulados por todos y cada uno de los miembros del equipo. Todos los requerimientos tienen atributos y estos son la principal fuente de información para ayudarle a planear, comunicar y rastrear las actividades del proyecto a través del ciclo de vida. Cada proyecto tiene necesidades únicas y se deberán seleccionar los atributos que sean críticos para asegurar su éxito: prioridad de desarrollo, status, autor, responsable, relaciones, fecha de registro, fecha última modificación, versión, etc.

RequisitePro permite la asignación de prioridades a los requerimientos en base a:

**Beneficios al cliente:** Todos los requerimientos no son desarrollados de igual forma. Se les da prioridad en base a la importancia relativa del usuario final basado en un análisis previo de los analistas y desarrolladores del equipo.

**Esfuerzo:** Claramente, algunos requerimientos o cambios a éstos demandan más tiempo y recursos que otros. Estimar el número de semanas-personas o líneas de código requeridas en base a requerimientos, es la mejor forma de determinar que y que no puede ser desarrollado en el tiempo estipulado.

Una característica importante es que la curva de aprendizaje de RequisitePro es pequeña, este aprendizaje puede ser basado en el uso de los tutoriales, los cuales guían por los puntos principales en el uso de la herramienta.

#### **Beneficios de RequisitePro**

- Permite el trabajo en equipo por medio de un repositorio compartido de información.
- Permite la clasificación de requerimientos, en base a las necesidades de cada empresa: usuario, técnicos, comunicación, pruebas.
- Define atributos para todos los tipos de requerimientos especificados.
- Ayuda a manipular el alcance del proyecto mediante la asignación de prioridad de desarrollo a cada uno de los requerimientos planteados.

- Características avanzadas de rastreo, por medio de matrices, que permiten visualizar las dependencias entre requerimientos dentro de un proyecto o en diferentes proyectos.
- Marcas que automáticamente indican cuándo un requerimiento es impactado por cambios a otro requerimiento o a atributos asociados.
- Administración de cambios mediante el rastreo y la visualización histórica de los cambios efectuados al requerimiento, cuándo y quién los realizó.
- Manejo de plantillas creadas por el usuario, o creadas por otras empresas.
- Recolección de requerimientos mediante su importación por medio de Wizards para obtenerlos automáticamente de fuentes externas, incluyendo archivos o bases de datos.
- Interactúa con los demás productos Rational para el ciclo de vida, así como con herramientas de Microsoft Office.
- Ayuda a determinar en forma automatizada cuántos requerimientos tiene el proyecto.
- Ayuda a determinar responsables y actores en cada uno de los requerimientos.
- RequisitePro, le permite organizar sus requerimientos, establecer y mantener relaciones padre/hijo entre ellos.

#### **4.9.2 Doors**

DOORS<sup>(R)</sup> es la herramienta de administración de requerimientos creada por Quality Systems and Software. Esta herramienta permite capturar, relacionar, analizar y administrar un rango de información para asegurar el cumplimiento del proyecto en materia de requerimientos.

DOORS' permite el acceso de un gran número de usuarios concurrentes en la red, manteniendo en línea un gran número de requerimientos así como su información asociada.

DOORS ayuda al usuario a procesar las solicitudes de cambios de requerimientos en línea. Permite realizar cualquier modificación vía remota cuando la base de datos está off-line, incorporando sus actualizaciones a la base de datos maestra.

Esto hace más fácil la comunicación del equipo con otras organizaciones, subcontratistas y proveedores.

Esta herramienta proporciona rastreabilidad multi-nivel para aquellas relaciones entre requerimientos que poseen gran tamaño. DOORS cuenta con un wizard que le permite generar enlaces a reportes de muchos niveles, para desplegarlos en la misma vista.

### **Beneficios de DOORS**

- Análisis y comparación de requerimientos.
- Clasificación de requerimientos.
- Interpretación manual de cada requerimiento.
- Identificación de Inconsistencias.
- Operación vía batch.
- Permite compartir requerimientos entre proyectos.
- Permite crear relaciones entre requerimientos mediante la táctica drag-and-drop
- Envía una notificación vía email cuando los cambios son revisados.
- Permite visualizar los cambios pendientes de otros usuarios para anticipar el impacto que ocasionará.
- Despliega estadísticas y métricas a través de gráficas.
- Los documentos están escritos en lenguaje claro, lo que proporciona una comprensión inmediata de cada requerimiento.
- Permite importar sus documentos a formatos de herramientas de Microsoft Office, RTF, HTML, texto, entre otros.
- Las plantillas presentan la información de manera estandarizada.

## **5. CONCLUSIONES**

- La ingeniería de requerimientos consta de cuatro etapas, estas se deben de desarrollar en cierto orden para conseguir los objetivos específicos de esta fase conformando así un proceso repetible, que requiere la aplicación de técnicas y herramientas que respalden dicho proceso con el objetivo de descubrir, analizar , especificar y validar requerimientos consistentes claros y apropiados .
- La elicitación de requerimientos es una etapa que representa el comienzo de la obtención de requerimientos y la conforman todas las actividades involucradas en el descubrimiento de los requerimientos del sistema, en este punto los analistas y el cliente deben trabajar juntos para poder definir el dominio del problema, los servicios que el sistema debe prestar, restricciones y demás características relevantes.
- Con respecto a las tareas, herramientas y técnicas de elicitación, se afirma que, en definitiva, si somos capaces de comprender lo que el cliente/usuario está diciendo, entonces estamos realizando nuestro trabajo en buena forma y como consecuencia estamos obteniendo buenos requerimientos.
- Las técnicas de elicitación de requerimientos permiten establecer un diálogo con el cliente/usuario; es decir, permiten el flujo de conocimientos entre las partes. En este sentido, se ha podido concluir que una de las mejores formas de comunicación que existen son las entrevistas; momentos en los que nos encontramos atendiendo al cliente, tratando de entender sus necesidades y problemas y viendo como podemos satisfacerlos y solucionarlos. Es imprescindible en estos momentos saber observar y escuchar.
- El análisis de requerimientos apunta descubrir problemas e inconsistencias entre los requerimientos, se procede a leer, conceptuar e investigar dichos requerimientos, se intercambian ideas con el equipo de trabajo, se resaltan problemas, se buscan soluciones y alternativas para luego presentar dicho estudio al cliente buscando su aprobación.



- En la documentación de requerimientos se pasa del lenguaje natural (comprendido por el cliente) al lenguaje formal y técnico, cada uno de los requerimientos obtenidos y analizados, haciendo uso de lenguajes y herramientas de índole formal o semi-formal, con el objetivo de facilitar la validación especializada de los requerimientos.
- En el presente trabajo se han definido numerosas técnicas y herramientas tanto de elicitación como de análisis de requerimientos pero de nada servirían si no existiese una metodología para aplicarlas, así se logra sacar el máximo provecho de ellas en las etapas que le correspondan según las tareas recomendadas de esta metodología propuesta. Asimismo, no basta con que utilicemos una sola herramienta para lograr nuestros fines: cada una de ellas aporta diferente y vital información sobre el problema en cuestión, todo depende de que tanto conocemos el dominio del problema y que tan complejo sea, para recurrir a las diferentes técnicas mencionadas anteriormente. De estas técnicas y herramientas fueron seleccionadas algunas de ellas para aplicarlas en la obtención de requerimientos de este proyecto generando resultados positivos en estado de aceptación.
- La mayoría de los proyectos de Tecnologías de Información y especialmente aquellos relacionados directamente con la producción de soluciones de software, no llegan a completarse con éxito, por causas tales como mala definición de requerimientos, requerimientos irreales, inconsistentes, dominio del problema desconocido o confuso y estimaciones desfasadas en tiempo, espacio y recursos, entre otras que para este caso no fueron tenidas en cuenta. Para aportar a la mitigación de errores en los sistemas producidos por las diferentes organizaciones especializadas en dicha área se debe de atacar cada una de las causas mencionadas anteriormente, para este fin se encontró que una adecuada obtención de requerimientos son soluciones eficientes en tales términos las cuales Deben de ser desarrolladas basándose en métodos que faciliten la consecución de sus objetivos.

## **6. RECOMENDACIONES**

El presente trabajo no abarco la metodología de la validación, seria recomendable continuar con la metodología de esta ultima actividad ya que es la actividad en donde se validan los requerimientos de almacenamiento de información, funcionales, no funcionales y se cierra la versión de los requerimientos, se cierran las versiones de los entregables o productos obtenidos para establecer un hito en el desarrollo, sin por ello dejar de tener en cuenta que probablemente serán necesarias futuras iteraciones del proceso a la luz de los resultados que se vayan obteniendo al avanzar en el desarrollo.

Se recomienda continuar con el estudio y análisis de un método optimo para la obtención de requerimientos, considerando que dichas fases son los cimientos principales y fundamentales para un desarrollo adecuado de software.

El método propuesto para la obtención de requerimientos puede ser mejorado aplicando otras técnicas y herramientas y estudiando los resultados obtenidos de dicha aplicación para de esta manera considerar diferentes formas de optimización, conformando así un método robusto y flexible.

Seria objeto de estudio la aplicación del método propuesto a más casos reales de desarrollo de software y estimar su validez a lo largo de dicho proceso.

## **7. REFERENCIAS**

### **REFERENCIAS BIBLIOGRAFICAS**

[SOMMERVILLE,1997] Requirements Engineering: A Good Practice guide. John Wiley & sons.

[POHL,1996] Pohl,K. Las tres Dimensiones de la ingeniería de requerimientos, un armazón y su aplicación , Los Sistemas de Información .

### **REFERENCIAS WEB**

[BAEZ et all ,2007]

[http://wer.inf.pucrio.br/WERpapers/artigos/artigos\\_WER01/baez.pdf](http://wer.inf.pucrio.br/WERpapers/artigos/artigos_WER01/baez.pdf)

Título: METODOLOGÍA DORCU PARA LA INGENIERÍA DE REQUERIMIENTOS

Autora: M. Grisela Báez ,Silvia I. Barba Brunner

Instituto Superior Politecnico “Jose Antonio Echevarria”, La Habana, CU

[DAVILA, 2007]

<http://www.monografias.com/trabajos12/ingreq/ingreq.shtml>

Título: UNA GUÍA PARA EXTRAER, ANALIZAR, ESPECIFICAR Y VALIDAR LOS REQUERIMIENTOS DE UN PROYECTO

Autor: Nicolás Davyt Dávila

Universidad de Uruguay

[SUMANO, 2006]

<http://www.geocities.com/diegolp/ingsof/requerimientos.pdf>

Título: ANÁLISIS DE REQUERIMIENTOS DE SOFTWARE. ESTADO DEL ARTE

Autor: M.C.C. María de los Ángeles Sumano López

Instituto Politécnico Nacional Centro De Investigación En Computación

[THOMAS, 2006]

[http://wer.inf.puc-rio.br/WERpapers/artigos/artigos\\_WER05/pablo\\_thomas.pdf](http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER05/pablo_thomas.pdf)

Título: ELICITACIÓN DE OBJETIVOS A PARTIR DE ESCENARIOS

Autor: Pablo Thomas

Instituto de Investigación en Informática LIDI

[ZAVALA, 2006]

<http://www.consol.org.mx/2004/material/63/por-que-fallan-los-proy-de-soft.pdf>

Título: ¿POR QUÉ FRACASAN LOS PROYECTOS DE SOFTWARE?; UN ENFOQUE ORGANIZACIONAL

Autor: J. Jesús María Zavala Ruiz

Universidad Autónoma Metropolitana-Iztapalapa Mexico, D.F.

[RANGEL, 2007]

[http://www.unipamplona.edu.co/KMportal/hermesoft/portallG/home\\_1/recursos/tesis/contenidos/pdf\\_tesis/02052007/analisis\\_comparativo.pdf](http://www.unipamplona.edu.co/KMportal/hermesoft/portallG/home_1/recursos/tesis/contenidos/pdf_tesis/02052007/analisis_comparativo.pdf)

Título: ANALISIS COMPARATIVO DE TÉCNICAS DE OBTENCIÓN DE REQUERIMIENTOS PARA EL MODULO DE FACTURACIÓN DEL APLICATIVO GESTASOFT HOSPITALARIO PARA IMSALUD

Autor: Adriana Milena Rangel Carrillo

Facultad de ingenierías y Arquitectura

## **8. ANEXOS**

### **8.1 Análisis comparativo de metodologías utilizadas para la obtención de requerimientos y la metodología propuesta**

<b>Metodología</b>	<b>Etapas</b>	<b>Ventajas</b>	<b>Desventajas</b>
<b>DoRCU (Documentación de requerimientos centrada en el usuario)</b>	<ol style="list-style-type: none"> <li>1. Captura de requerimientos</li> <li>2. Análisis de Requerimientos</li> <li>3. Especificación de requerimientos</li> <li>4. Validación y Certificación de Requerimientos</li> </ol>	<p>Metodología basada en el usuario la cual permite satisfacer las necesidades reales del cliente.</p> <p>Documentación de requerimientos de una manera técnica para el equipo de desarrollo y en lenguaje natural para el usuario facilitando su comprensión para la revisión del usuario .</p> <p>Permite el desarrollo en ciclos o iteraciones para de esta manera corregir o adicionar requerimientos.</p>	<p>Se debe dedicar bastante tiempo para la realización de los documentos lo cual retrasa en parte al desarrollo del proyecto</p> <p>Duplicidad de trabajo por la necesidad de desarrollar documentos de manera formal para el equipo de trabajo y pasarlo de una manera informal para el usuario</p>
<b>GBRAM(Goal Based Requirements Analysis Method)</b>	<ol style="list-style-type: none"> <li>1. Identificar metas y objetivos</li> <li>2. Organizar y clasificar metas</li> <li>3. Refinar y elaborar metas</li> <li>4. Operacionalizar metas en requerimientos</li> </ol>	<p>Provee mecanismos de representación adecuados como lo son los escenarios para la comprensión de los stakeholders</p> <p>Se concentra en establecer los fundamentos que justifican los requerimientos de software debido a que se basa en los objetivos de la organización.</p>	<p>Su enfoque en escenario provee flexibilidad para obtener los requerimientos pero a su vez le resta formalismo a su especificación causando de esta manera conflictos en el equipo de desarrollo debido a posible ambigüedades en los requerimientos</p> <p>No se define actividades para la especificación y validación de requerimientos</p>
<b>SSM (Soft System Methodology)</b>	<ol style="list-style-type: none"> <li>1. Poner en claro la situación del problema</li> <li>2. Expresar situación</li> </ol>	<p>Se enfoca en el modelamiento visual lo que permite una mayor comprensión</p>	<p>Es recomendable para la comprensión del ambiente del proyecto y el dominio</p>

**DESARROLLO DE UNA METODOLOGÍA PARA LA ELICITACIÓN Y ANÁLISIS DEL PROCESO DE LA INGENIERÍA DE REQUERIMIENTOS**

	<ol style="list-style-type: none"> <li>Hacer la selección de cómo ver paulatinamente la situación de manera que se pueda producir percepciones que den pie a definiciones raíz</li> <li>Construir modelos conceptuales</li> <li>Comparar los modelos conceptuales con el mundo real</li> <li>Identificar cambios factibles y deseables</li> <li>Recomendaciones para la toma de acciones que mejoren la percepción del problema</li> </ol>	<p>de los requerimientos</p> <p>Esta basado en la representación de actividades del mundo real</p> <p>Permite obtener la información inicial</p>	<p>del problema pero no para utilizarla como metodología de requerimientos en un proyecto formal debido a q no proporciona la información necesaria para el desarrollo de las actividades de dicha fase</p>
<b>Metodología Propuesta</b>	<ol style="list-style-type: none"> <li>Elicitación de requerimientos</li> <li>Análisis de Requerimientos</li> <li>Documentación de Requerimientos</li> <li>Validación de Requerimientos</li> </ol>	<p>de</p> <p>Metodología basada en los clientes, desarrolladores, analistas y demás involucrados satisfaciendo así las necesidades reales del cliente y obteniendo buenos requerimientos ya que el cliente participa en cada actividad del proceso para la revisión y conformidad de cada requerimiento</p> <p>Tiene como producto final un Documento de requerimientos que contiene todas las plantillas, modelos o prototipos que se definieron en cada actividad.</p> <p>Permite el desarrollo en ciclos de acuerdo al modelo propuesto y de esta manera se puede retroceder a una actividad anterior y asegurarse de especificar buenos requerimientos</p>	<p>Es recomendable analizar mas las técnicas para los requerimientos no funcionales y continuar con la metodología de la validación que no esta definida.</p>

**Tabla 8.1:**      **Análisis comparativo de algunas metodologías de obtención de requerimientos**

## **8.2 Factores críticos de éxito del Proyecto**

Para llevar a cabo con éxito el desarrollo del sistema software se deben de orientar los mejores recursos en la realización de los factores claves de éxito, que de lograrse permite alcanzar con éxito los objetivos proyecto a realizar.

Los factores críticos de Éxito más importantes son los siguientes:

- Compromiso de los stakeholder, una de las primeras acciones que se debe realizar, de manera que, apoyen al equipo del proyecto en todo lo necesario.
- Participación de los usuarios debido a su experiencia directa e indirecta, permite que conozcan las fallas y procesos inadecuados; también pueden participar activamente en la identificación de las necesidades de información no satisfechas.
- Alcance claramente definido, es necesario que el grupo de técnicos (analistas, ingenieros, etc.) tenga el conocimiento y concepción clara de los objetivos, actividades y productos a obtener, así como también de las técnicas requeridas en la elaboración del Plan de Sistemas.
- Organización y experiencia del grupo de trabajo, el equipo de trabajo debe de poseer los conocimientos en: tecnología de la información, así como la experiencia suficiente para poder interpretar los requerimientos funcionales y poder traducirlos en aplicaciones y/o sistemas de información.

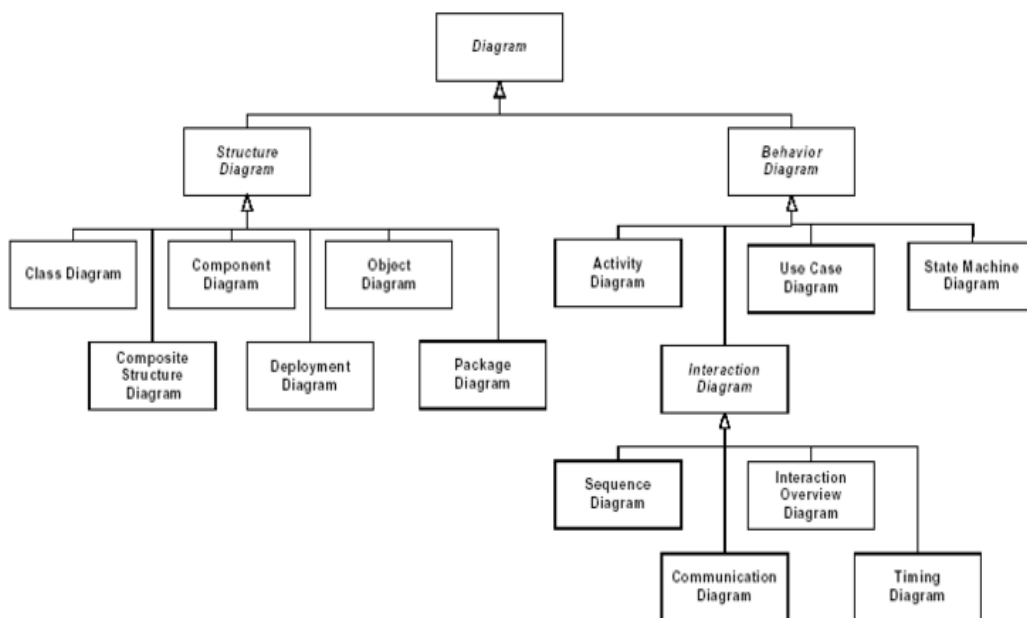
## **8.3 UML**

A partir del año 1994, Grady Booch [Booch96](precursor de Booch '93) y Jim Rumbaugh (creador de OMT) se unen en una empresa común, Rational Software Corporation, y comienzan a unificar sus dos métodos. Un año más tarde, en octubre de 1995, aparece UML (Unified Modeling Language) 0.8, la que se considera como

la primera versión del UML. A finales de ese mismo año, Ivan Jacobson, creador de OOSE (Object Oriented Software Engineer) se añade al grupo.

Como objetivos principales de la consecución de un nuevo método que aunara los mejores aspectos de sus predecesores, sus protagonistas se propusieron lo siguiente:

- El método debía ser capaz de modelar no sólo sistemas de software sino otro tipo de sistemas reales de la empresa, siempre utilizando los conceptos de la orientación a objetos (OO).
- Crear un lenguaje para modelado utilizable a la vez por máquinas y por personas.
- Establecer un acoplamiento explícito de los conceptos y los artefactos ejecutables.
- Manejar los problemas típicos de los sistemas complejos de misión crítica.



**Figura 8.1:** Jerarquía de los diagramas UML 2.0



En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente, como se muestra en la figura de la derecha.

**Diagramas de estructura** enfatizan en los elementos que deben existir en el sistema modelado:

- Diagrama de clases
- Diagrama de componentes
- Diagrama de objetos
- Diagrama de estructura compuesta (UML 2.0)
- Diagrama de despliegue
- Diagrama de paquetes

**Diagramas de comportamiento** enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades
- Diagrama de casos de uso
- Diagrama de estados

**Diagramas de Interacción**, un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia
- Diagrama de comunicación
- Diagrama de tiempos (UML 2.0)
- Diagrama de vista de interacción (UML 2.0)

#### **4.10 Lenguaje restrictivo (constraint) de objetos (OCL)**

Una imagen puede describir muchas palabras. De igual modo, un modelo gráfico puede describir una cierta parte del comportamiento, después de la cual es

necesario rellenar detalles adicionales con palabras. Describiendo algo con palabras, sin embargo, casi siempre desemboca en ambigüedades; por ejemplo, "¿que quería decir cuando escribió eso?". El Lenguaje Restringido (constraint) de Objetos (OCL) está incorporado en UML como un estándar para especificar detalles adicionales, o precisar detalles en la estructura de los modelos.

Desarrollado dentro de la IBM Insurance Division como un lenguaje de modelado de negocio, el OCL es un lenguaje formal diseñado para ser fácil de leer y de escribir. OCL es más funcional que el lenguaje natural, pero no tan preciso como un lenguaje de programación - no puede ser usado para escribir lógicas de lógica de programación o control de flujo. Puesto que OCL es un lenguaje para la expresión pura, sus declaraciones están garantizadas de no tener efectos laterales - simplemente transportan un valor y nunca pueden cambiar el estado del sistema.